

# Multitask diffusion adaptation over asynchronous networks

Roula Nassif, *Student Member, IEEE*, Cédric Richard, *Senior Member, IEEE*  
 André Ferrari, *Member, IEEE*, Ali H. Sayed, *Fellow Member, IEEE*

**Abstract**—The multitask diffusion LMS is an efficient strategy to simultaneously infer, in a collaborative manner, multiple parameter vectors. Existing works on multitask problems assume that all agents respond to data synchronously. In several applications, agents may not be able to act synchronously because networks can be subject to several sources of uncertainties such as changing topology, random link failures, or agents turning on and off for energy conservation. In this work, we describe a model for the solution of multitask problems over asynchronous networks and carry out a detailed mean and mean-square error analysis. Results show that sufficiently small step-sizes can still ensure both stability and performance. Simulations and illustrative examples are provided to verify the theoretical findings.

**Index Terms**—Distributed optimization, asynchronous networks, diffusion adaptation, multitask learning, mean-square performance analysis.

## I. INTRODUCTION

Distributed adaptive learning enables agents to learn a concept via local information exchange, and to continuously adapt to track possible concept drifts. Distributed implementations offer an attractive alternative to centralized solutions with advantages related to scalability, robustness, and decentralization (see, e.g., [2], [3] and the many examples therein). Several strategies for distributed online parameter estimation have been proposed in the literature, including consensus strategies [4]–[9], incremental strategies [10]–[14], and diffusion strategies [15]–[20]. Incremental techniques operate on a cyclic path that runs across all nodes, which makes them sensitive to link failures and problematic for adaptive implementations. On the other hand, diffusion strategies are particularly attractive due to their enhanced adaptation performance and wider stability ranges than consensus-based implementations. Accessible overviews of results on diffusion adaptation can be found in [2], [15], [16].

Most prior literature focuses primarily on the case where nodes estimate a single parameter vector collaboratively. We refer to problems of this type as single-task problems. Some applications require more complex models and flexible algorithms than single-task implementations since their agents may

involve the need to track multiple targets simultaneously. For instance, sensor networks deployed to estimate a spatially-varying temperature profile need to exploit more directly the spatio-temporal correlations that exist between measurements at neighboring nodes [21]. Likewise, monitoring applications where agents need to track the movement of multiple correlated targets need to exploit the correlation profile in the data for enhanced accuracy. Problems of this kind, where nodes need to infer multiple parameter vectors, are referred to as multitask problems.

Existing strategies to address multitask problems mostly depend on how the tasks relate to each other and on exploiting some prior information. There have been some useful works dealing with such problems over distributed networks. For example, in [22] a diffusion strategy of the LMS type is developed to solve distributed optimization problems where nodes are interested in estimating parameters of local interest and parameters of global interest to the whole network. In [23], an extension of the diffusion algorithm developed in [22] allows nodes to estimate parameters of common interest to a subset of nodes simultaneously with parameters of local and global interest. In comparison, the parameter space is decomposed into two orthogonal subspaces in [24], with one of the subspaces being common to all nodes. Multitask estimation algorithms over fully connected networks and tree networks are also considered in [25], [26]. These works assume that the node-specific parameter vectors lie in a common latent signal subspace and exploit this property to compress information and to reduce communication costs. An alternative way to exploit and model relationships among tasks is to formulate optimization problems with appropriate co-regularizers between nodes [27], [28]. The multitask diffusion LMS algorithm derived in [27] relies on this principle, and we build on this construction in this article. In this context, the network is not assumed to be fully connected and agents need not be interested in some common parameters. It is sufficient to assume that different clusters within the network are interested in their own models, and that there are some correlations among the models of adjacent clusters. These correlations are captured by means of regularization parameters. Multitask estimation problems have also been addressed over diffusion networks where no prior information on possible relationships between tasks is assumed and nodes do not know which other nodes share the same task [29]–[32]. In this case, it was argued in [29] that the diffusion iterates converge to a Pareto optimal solution when confronted with a multi-objective optimization problem. To avoid cooperation between

The work of C. Richard and A. Ferrari was partly supported by ANR and DGA grant ANR-13-ASTR-0030 (ODISSEE project). The work of A. H. Sayed was supported in part by NSF grants CIF-1524250 and ECCS-1407712. A short version of this work appears in the conference publication [1].

R. Nassif, C. Richard, and A. Ferrari are with the Université de Nice Sophia-Antipolis, France (email: roula.nassif@oca.eu; cedric.richard@unice.fr; andre.ferrari@unice.fr).

A. H. Sayed is with the department of electrical engineering, University of California, Los Angeles, USA (email: sayed@ee.ucla.edu).

neighbors seeking different objectives, automatic clustering techniques using diffusion strategies have been proposed. The clustering techniques developed in [30], [31] are based on setting the combination coefficients in an online manner. The technique proposed in [33] is based on solving a hypothesis test problem for setting the neighborhood in an online manner.

The aforementioned works on multitask problems assume that all agents respond to data synchronously. In several applications, agents may not be able to act synchronously because networks can be subject to several sources of uncertainties such as changing topology, random link failures, or agents turning on and off. There exist several useful studies in the literature on the performance of consensus and gossip strategies in the presence of asynchronous events [8], [9], [34], [35] or changing topologies [8], [9], [35]–[41]. In most parts, these works investigate pure averaging algorithms that cannot process streaming data or the works assume noise-free data or make use of decreasing step-size sequences. There are also studies in the context of diffusion strategies. In particular, the works [42]–[44] advanced a rather general framework for asynchronous networks that includes many prior models as special cases. The works examined how asynchronous events interfere with the behavior of adaptive networks in the presence of streaming noisy data and under constant step-size adaptation. Several interesting conclusions are reported in [44] where comparisons are carried out between synchronous and asynchronous behavior, as well as with centralized solutions. In the current work, we would like to examine similar effects to [42], [43] albeit in the context of multitask networks as opposed to single-task networks. In this case, a new dimension arises in that asynchronous events can interfere with the exchange of information among clusters. We examine in some detail the mean and mean-square stability of the multitask network and show that sufficiently small step-sizes can still ensure convergence and performance. Various simulation results illustrate the theoretical findings.

This paper is organized as follows. In Section II, we briefly recall the multitask diffusion LMS strategy and we introduce a fairly general model for asynchronous behavior. Under this model, agents in the network may stop updating their solutions, or may stop sending or receiving information in a random manner. Section III analyzes the theoretical performance of the algorithm, in the mean and mean-square error sense. In Section IV, experiments are presented to illustrate the performance of the diffusion multitask approach over asynchronous networks.

## II. MULTITASK DIFFUSION LMS OVER ASYNCHRONOUS NETWORKS

Before starting our presentation, we provide a summary of some of the main symbols used in the article. Other symbols will be defined in the context where they are used:

$x$	Normal font letters denote scalars.
$\mathbf{x}$	Boldface lowercase letters denote column vectors.
$\mathbf{R}$	Boldface uppercase letters denote matrices.
$(\cdot)^\top$	Matrix transpose.
$(\cdot)^{-1}$	Matrix inverse.
$\mathbf{I}_N$	Identity matrix of size $N \times N$ .
$\mathcal{N}_k$	The set of nodes containing the neighborhood of node $k$ , including $k$ .
$\mathcal{N}_k^-$	The set of nodes containing the neighborhood of node $k$ , excluding $k$ .
$\mathcal{C}_j$	Cluster $j$ , i.e., index set of nodes in the $j$ -th cluster.
$\mathcal{C}(k)$	The cluster of nodes to which node $k$ belongs, including $k$ .
$\mathcal{C}(k)^-$	The cluster of nodes to which node $k$ belongs, excluding $k$ .

We now briefly recall the *synchronous* diffusion adaptation strategy developed in [27] for solving distributed optimization problems over multitask networks.

### A. Multitask diffusion adaptation

We consider a connected network consisting of  $N$  nodes grouped into  $Q$  clusters, as illustrated in Figure 1. The problem is to estimate an  $L \times 1$  unknown vector  $\mathbf{w}_k^*$  at each node  $k$  from collected data. Node  $k$  has access to temporal measurement sequences  $\{d_k(i), \mathbf{x}_k(i)\}$ , where  $d_k(i)$  is a scalar zero-mean reference signal, and  $\mathbf{x}_k(i)$  is an  $L \times 1$  regression vector with a positive-definite covariance matrix  $\mathbf{R}_{x,k} = E\{\mathbf{x}_k(i)\mathbf{x}_k^\top(i)\} > 0$ . The data at node  $k$  are assumed to be related via the linear regression model

$$d_k(i) = \mathbf{x}_k^\top(i) \mathbf{w}_k^* + z_k(i), \quad (1)$$

where  $z_k(i)$  is a zero-mean i.i.d. noise of variance  $\sigma_{z,k}^2$  that is independent of any other signal. We assume that nodes belonging to the same cluster have the same parameter vector to estimate, namely,

$$\mathbf{w}_k^* = \mathbf{w}_{\mathcal{C}_q}^*, \quad \text{whenever } k \in \mathcal{C}_q. \quad (2)$$

We say that two clusters are connected if there exists at least one edge linking a node from one cluster to a node in the other cluster. We also assume that relationships between connected clusters exist so that cooperation among adjacent clusters is beneficial. In particular, we suppose that the parameter vectors corresponding to two connected clusters  $\mathcal{C}_p$  and  $\mathcal{C}_q$  satisfy certain properties, such as being close to each other [27]. Cooperation across these clusters can therefore be beneficial to infer  $\mathbf{w}_{\mathcal{C}_p}^*$  and  $\mathbf{w}_{\mathcal{C}_q}^*$ .

Consider the cluster  $\mathcal{C}(k)$  to which node  $k$  belongs. A local cost function,  $J_k(\mathbf{w}_{\mathcal{C}(k)})$ , is associated with node  $k$ . It is assumed to be strongly convex and second-order differentiable, an example of which is the mean-square error criterion considered throughout this paper and defined by

$$J_k(\mathbf{w}_{\mathcal{C}(k)}) = \mathbb{E}\{|d_k(i) - \mathbf{x}_k^\top(i) \mathbf{w}_{\mathcal{C}(k)}|^2\}. \quad (3)$$

Depending on the application, there may be certain properties among the optimal vectors  $\{\mathbf{w}_{\mathcal{C}_1}^*, \dots, \mathbf{w}_{\mathcal{C}_Q}^*\}$  that deserve to be promoted in order to enhance estimation accuracy. Among



links are used to promote relationships between tasks. In a manner similar to [42], the asynchronous network model is assumed to satisfy the following conditions:

- *Conditions on the step-size parameters:* At each time instant  $i$ , the step-size at node  $k$  is a bounded nonnegative random variable  $\mu_k(i) \in [0, \mu_{\max, k}]$ . These step-sizes are collected into the random matrix  $\mathbf{M}(i) \triangleq \text{diag}\{\mu_1(i), \dots, \mu_N(i)\}$ . We assume that  $\{\mathbf{M}(i), i \geq 0\}$  is a weakly stationary random process with mean  $\bar{\mathbf{M}}$  and Kronecker-covariance matrix  $\mathbf{C}_M$  of size  $N^2 \times N^2$  defined as

$$\mathbf{C}_M \triangleq \mathbb{E}\{(\mathbf{M}(i) - \bar{\mathbf{M}}) \otimes (\mathbf{M}(i) - \bar{\mathbf{M}})\} \quad (10)$$

with  $\otimes$  denoting the Kronecker product.

- *Conditions on the combination coefficients:* The random coefficients  $\{a_{\ell k}(i)\}$  used to scale the estimates  $\{\psi_\ell(i+1)\}$  that are being received by node  $k$  from its cluster neighbors  $\ell \in \mathcal{N}_k(i) \cap \mathcal{C}(k)$  satisfy the following constraints at each iteration  $i$ :

$$\sum_{\ell \in \mathcal{N}_k(i) \cap \mathcal{C}(k)} a_{\ell k}(i) = 1, \text{ and } \begin{cases} a_{\ell k}(i) > 0, & \text{if } \ell \in \mathcal{N}_k(i) \cap \mathcal{C}(k) \\ a_{\ell k}(i) = 0, & \text{otherwise.} \end{cases} \quad (11)$$

We collect these coefficients into the random  $N \times N$  left-stochastic matrix  $\mathbf{A}(i)$ . We again assume that  $\{\mathbf{A}(i), i \geq 0\}$  is a weakly stationary random process. Let  $\bar{\mathbf{A}}$  be its mean and  $\mathbf{C}_A$  its Kronecker-covariance matrix of size  $N^2 \times N^2$  defined as

$$\mathbf{C}_A \triangleq \mathbb{E}\{(\mathbf{A}(i) - \bar{\mathbf{A}}) \otimes (\mathbf{A}(i) - \bar{\mathbf{A}})\}. \quad (12)$$

- *Conditions on the regularization factors:* The random factors  $\{\rho_{k\ell}(i)\}$ , which adjust the regularization strength between the parameter vectors at neighboring nodes of distinct clusters, satisfy the following constraints at each iteration  $i$ :

$$\sum_{\ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k)} \rho_{k\ell}(i) = 1, \text{ and } \begin{cases} \rho_{k\ell}(i) > 0, & \text{if } \ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k) \\ \rho_{kk}(i) \geq 0, \\ \rho_{k\ell}(i) = 0, & \text{otherwise.} \end{cases} \quad (13)$$

We collect these coefficients into the random  $N \times N$  right-stochastic matrix  $\mathbf{P}(i)$ . We assume that  $\{\mathbf{P}(i), i \geq 0\}$  is a weakly stationary random process with mean  $\bar{\mathbf{P}}$  and Kronecker-covariance matrix  $\mathbf{C}_P$  of size  $N^2 \times N^2$  defined as

$$\mathbf{C}_P \triangleq \mathbb{E}\{(\mathbf{P}(i) - \bar{\mathbf{P}}) \otimes (\mathbf{P}(i) - \bar{\mathbf{P}})\}. \quad (14)$$

- *Independence assumptions:* To enable tractable analysis, we shall assume that the random matrices  $\mathbf{M}(i)$ ,  $\mathbf{A}(i)$ , and  $\mathbf{P}(i)$  at iteration  $i$  are mutually-independent and independent of any other random variables. These matrices are related to node, intra-cluster and inter-cluster link failures, respectively.
- *Mean graph:* The mean matrices  $\bar{\mathbf{A}}$  and  $\bar{\mathbf{P}}$  define the intra-cluster and inter-cluster neighborhoods, namely,  $\mathcal{N}_k \cap \mathcal{C}(k)$  and  $\mathcal{N}_k \setminus \mathcal{C}(k)$  for all  $k$ , respectively. We refer to the neighborhoods  $\mathcal{N}_k = (\mathcal{N}_k \cap \mathcal{C}(k)) \cup (\mathcal{N}_k \setminus \mathcal{C}(k))$  for all  $k$ , defined by  $\bar{\mathbf{A}}$  and  $\bar{\mathbf{P}}$ , as the mean graph. In view of the above conditions, the mean combination

coefficients  $\bar{a}_{\ell k} \triangleq \mathbb{E}\{a_{\ell k}(i)\}$  and regularization factors  $\bar{\rho}_{k\ell} \triangleq \mathbb{E}\{\rho_{k\ell}(i)\}$  are nonnegative and satisfy the following constraints.

$$\sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} \bar{a}_{\ell k} = 1, \text{ and } \begin{cases} \bar{a}_{\ell k} > 0, & \text{if } \ell \in \mathcal{N}_k \cap \mathcal{C}(k), \\ \bar{a}_{\ell k} = 0, & \text{otherwise,} \end{cases} \quad (15)$$

$$\sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \bar{\rho}_{k\ell} = 1, \text{ and } \begin{cases} \bar{\rho}_{k\ell} > 0, & \text{if } \ell \in \mathcal{N}_k \setminus \mathcal{C}(k), \\ \bar{\rho}_{kk} \geq 0, \\ \bar{\rho}_{k\ell} = 0, & \text{otherwise.} \end{cases} \quad (16)$$

Using the same arguments as Lemmas 2 and 3 in [42], we can state the following properties for the asynchronous model (9).

**Property 1.** The  $N \times N$  matrix  $\bar{\mathbf{A}}$  and the  $N^2 \times N^2$  matrix  $\bar{\mathbf{A}} \otimes \bar{\mathbf{A}} + \mathbf{C}_A$  are left-stochastic matrices.

**Property 2.** The  $N \times N$  matrix  $\bar{\mathbf{P}}$  and the  $N^2 \times N^2$  matrix  $\bar{\mathbf{P}} \otimes \bar{\mathbf{P}} + \mathbf{C}_P$  are right-stochastic matrices.

**Property 3.** For every node  $k$ , the neighborhood  $\mathcal{N}_k$  that is defined by the mean graph of the asynchronous model (9) is equal to the union of all possible realizations for the random neighborhood  $\mathcal{N}_k(i) = (\mathcal{N}_k(i) \cap \mathcal{C}(k)) \cup (\mathcal{N}_k(i) \setminus \mathcal{C}(k))$ .

We provide in Appendix A one example for a common asynchronous network referred to as the Bernoulli network. The Bernoulli model proposed in [42] is more general than the one used for modeling random link failures in consensus networks [8], [37] since it also allows to consider random “on-off” behavior for agents. When dealing with multitask problems over asynchronous network, additional sources of uncertainties must be considered. The network provided in Appendix A allows us to jointly model intra-cluster link failures, inter-cluster link failures, and random “on-off” behaviors for agents.

### III. PERFORMANCE OF MULTITASK DIFFUSION OVER ASYNCHRONOUS NETWORKS

The performance of the multitask diffusion algorithm (9) is affected by various random perturbations due to the asynchronous events. We now examine the stochastic behavior of this strategy in the mean and mean-square error sense.

#### A. Mean error behavior analysis

For each agent  $k$ , we introduce the weight error vectors:

$$\tilde{\mathbf{w}}_k(i) \triangleq \mathbf{w}_k^* - \mathbf{w}_k(i), \quad \tilde{\boldsymbol{\psi}}_k(i) \triangleq \mathbf{w}_k^* - \boldsymbol{\psi}_k(i) \quad (17)$$

where  $\mathbf{w}_k^*$  is the optimum parameter vector at node  $k$ . We denote by  $\tilde{\mathbf{w}}(i)$ ,  $\tilde{\boldsymbol{\psi}}(i)$ , and  $\mathbf{w}^*$  the block weight error vector, the block intermediate weight error vector, and the block optimum weight vector, all of size  $N \times 1$  with blocks of size  $L \times 1$ , namely,

$$\tilde{\mathbf{w}}(i) \triangleq \text{col}\{\tilde{\mathbf{w}}_1(i), \dots, \tilde{\mathbf{w}}_N(i)\} \quad (18)$$

$$\tilde{\boldsymbol{\psi}}(i) \triangleq \text{col}\{\tilde{\boldsymbol{\psi}}_1(i), \dots, \tilde{\boldsymbol{\psi}}_N(i)\} \quad (19)$$

$$\mathbf{w}^* \triangleq \text{col}\{\mathbf{w}_1^*, \dots, \mathbf{w}_N^*\}. \quad (20)$$

We also introduce the following  $N \times N$  block matrices with individual entries of size  $L \times L$ :

$$\mathcal{M}(i) \triangleq \mathbf{M}(i) \otimes \mathbf{I}_L \quad (21)$$

$$\mathcal{A}(i) \triangleq \mathbf{A}(i) \otimes \mathbf{I}_L \quad (22)$$

$$\mathcal{P}(i) \triangleq \mathbf{P}(i) \otimes \mathbf{I}_L. \quad (23)$$

To perform the theoretical analysis, we introduce the following independence assumption.

**Assumption 1.** (*Independent regressors*) The regression vectors  $\mathbf{x}_k(i)$  arise from a stationary random process that is temporally stationary, temporally white, and independent over space with  $\mathbf{R}_{x,k} = E\{\mathbf{x}_k(i)\mathbf{x}_k^\top(i)\} > 0$ .

A direct consequence is that  $\mathbf{x}_k(i)$  is independent of  $\tilde{\mathbf{w}}_\ell(j)$  for all  $\ell$  and  $j \leq i$ . Although not true in general, this assumption is commonly used to analyze adaptive constructions since it allows to simplify the derivations without constraining the conclusions. There are several results in the adaptation literature that show that performance results that are obtained under the above independence assumptions match well the actual performance of the algorithms when the step-sizes are sufficiently small (see, e.g., [48, App. 24.A] and the many references therein).

The estimation error in the first step of the asynchronous strategy (9) can be rewritten as:

$$d_k(i) - \mathbf{x}_k^\top(i)\mathbf{w}_k(i) = \mathbf{x}_k^\top(i)\tilde{\mathbf{w}}_k(i) + z_k(i). \quad (24)$$

Subtracting  $\mathbf{w}_k^*$  from both sides of the adaptation step in (9) and using the above relation, we can express the update equation for  $\tilde{\mathbf{w}}(i+1)$  as:

$$\tilde{\mathbf{w}}(i+1) = [\mathbf{I}_{NL} - \mathcal{M}(i)(\mathbf{R}_x(i) + \eta \mathcal{Q}(i))]\tilde{\mathbf{w}}(i) - \mathcal{M}(i)\mathbf{p}_{xz}(i) + \eta \mathcal{M}(i)\mathcal{Q}(i)\mathbf{w}^* \quad (25)$$

where

$$\mathcal{Q}(i) \triangleq \mathbf{I}_{NL} - \mathcal{P}(i), \quad (26)$$

while  $\mathbf{R}_x(i)$  is an  $N \times N$  block matrix with individual entries of size  $L \times L$  given by

$$\mathbf{R}_x(i) \triangleq \text{diag}\{\mathbf{x}_1(i)\mathbf{x}_1^\top(i), \dots, \mathbf{x}_N(i)\mathbf{x}_N^\top(i)\}, \quad (27)$$

and  $\mathbf{p}_{xz}(i)$  is the  $N \times 1$  block column vector with blocks of size  $L \times 1$  defined as

$$\mathbf{p}_{xz}(i) \triangleq \text{col}\{\mathbf{x}_1(i)z_1(i), \dots, \mathbf{x}_N(i)z_N(i)\}. \quad (28)$$

Subtracting  $\mathbf{w}_k^*$  from both sides of the combination step in (9), we get the block weight error vector:

$$\tilde{\mathbf{w}}(i+1) = \mathcal{A}^\top(i)\tilde{\mathbf{w}}(i+1). \quad (29)$$

Substituting (25) into (29) we find that the error dynamics of the asynchronous multitask diffusion strategy (9) evolves according to the following recursion:

$$\begin{aligned} \tilde{\mathbf{w}}(i+1) &= \mathcal{A}^\top(i)[\mathbf{I}_{NL} - \mathcal{M}(i)(\mathbf{R}_x(i) + \eta \mathcal{Q}(i))]\tilde{\mathbf{w}}(i) - \\ &\quad \mathcal{A}^\top(i)\mathcal{M}(i)\mathbf{p}_{xz}(i) + \eta \mathcal{A}^\top(i)\mathcal{M}(i)\mathcal{Q}(i)\mathbf{w}^*. \end{aligned} \quad (30)$$

For compactness of notation, we introduce the symbols:

$$\mathcal{B}(i) \triangleq \mathcal{A}^\top(i)[\mathbf{I}_{NL} - \mathcal{M}(i)(\mathbf{R}_x(i) + \eta \mathcal{Q}(i))] \quad (31)$$

$$\mathbf{g}(i) \triangleq \mathcal{A}^\top(i)\mathcal{M}(i)\mathbf{p}_{xz}(i) \quad (32)$$

$$\mathbf{r}(i) \triangleq \mathcal{A}^\top(i)\mathcal{M}(i)\mathcal{Q}(i)\mathbf{w}^*, \quad (33)$$

so that (30) can be written as

$$\tilde{\mathbf{w}}(i+1) = \mathcal{B}(i)\tilde{\mathbf{w}}(i) - \mathbf{g}(i) + \eta \mathbf{r}(i). \quad (34)$$

Taking the expectation of both sides, using Assumption 1, and the independence of  $\mathbf{A}(i)$ ,  $\mathbf{M}(i)$ , and  $\mathbf{P}(i)$ , the network mean error vector ends up evolving according to the following dynamics:

$$\mathbb{E}\{\tilde{\mathbf{w}}(i+1)\} = \mathcal{B}\mathbb{E}\{\tilde{\mathbf{w}}(i)\} + \eta \mathbf{r} \quad (35)$$

where

$$\mathcal{B} \triangleq \mathbb{E}\{\mathcal{B}(i)\} = \mathcal{A}^\top[\mathbf{I}_{NL} - \mathcal{M}(\mathbf{R}_x + \eta \mathcal{Q})] \quad (36)$$

$$\mathbf{r} \triangleq \mathbb{E}\{\mathbf{r}(i)\} = \mathcal{A}^\top \mathcal{M} \mathcal{Q} \mathbf{w}^*, \quad (37)$$

where  $\mathcal{A}$ ,  $\mathcal{M}$ ,  $\mathbf{R}_x$ , and  $\mathcal{Q}$  denote the expectations of  $\mathcal{A}(i)$ ,  $\mathcal{M}(i)$ ,  $\mathbf{R}_x(i)$ , and  $\mathcal{Q}(i)$ , respectively, and are given by:

$$\mathcal{A} \triangleq \mathbb{E}\{\mathcal{A}(i)\} = \bar{\mathbf{A}} \otimes \mathbf{I}_L \quad (38)$$

$$\mathcal{M} \triangleq \mathbb{E}\{\mathcal{M}(i)\} = \bar{\mathbf{M}} \otimes \mathbf{I}_L \quad (39)$$

$$\mathcal{P} \triangleq \mathbb{E}\{\mathcal{P}(i)\} = \bar{\mathbf{P}} \otimes \mathbf{I}_L \quad (40)$$

$$\mathbf{R}_x \triangleq \mathbb{E}\{\mathbf{R}_x(i)\} = \text{diag}\{\mathbf{R}_{x,1}, \dots, \mathbf{R}_{x,N}\} \quad (41)$$

$$\mathcal{Q} \triangleq \mathbb{E}\{\mathcal{Q}(i)\} = \mathbf{I}_{NL} - \mathbb{E}\{\mathcal{P}(i)\} = \mathbf{I}_{NL} - \mathcal{P}. \quad (42)$$

Note that  $\mathbb{E}\{\mathbf{g}(i)\} = 0$  since  $z_k(i)$  is zero-mean and independent of any other signal.

**Theorem 1. (Stability in the mean)** Assume data model (1) and Assumption 1 hold. Then, for any initial condition, the multitask diffusion LMS strategy (9) applied to asynchronous networks converges asymptotically in the mean if, and only if, the step-sizes in  $\mathcal{M}$  are chosen to satisfy

$$\rho(\mathcal{A}^\top[\mathbf{I}_{NL} - \mathcal{M}(\mathbf{R}_x + \eta \mathcal{Q})]) < 1, \quad (43)$$

where  $\rho(\cdot)$  denotes the spectral radius of its matrix argument. In that case, the asymptotic mean bias is given by

$$\lim_{i \rightarrow \infty} \mathbb{E}\{\tilde{\mathbf{w}}(i)\} = \eta (\mathbf{I}_{NL} - \mathcal{B})^{-1} \mathbf{r}. \quad (44)$$

Assume that the expected values for all step-sizes are uniform, namely,  $\mathbb{E}\{\mu_k(i)\} = \bar{\mu}$  for all  $k$ . A sufficient condition for (43) to hold is to ensure that

$$0 < \bar{\mu} < \frac{2}{\max_{1 \leq k \leq N} \rho(\mathbf{R}_{x,k}) + 2\eta}. \quad (45)$$

*Proof:* Convergence in the mean requires the matrix  $\mathcal{B}$  in (35) to be stable. Since any induced matrix norm is lower bounded by its spectral radius, we can write in terms of the block maximum norm [16]:

$$\begin{aligned} &\rho(\mathcal{A}^\top[\mathbf{I}_{NL} - \mathcal{M}(\mathbf{R}_x + \eta \mathcal{Q})]) \\ &\leq \|\mathcal{A}^\top[\mathbf{I}_{NL} - \mathcal{M}(\mathbf{R}_x + \eta \mathcal{Q})]\|_{b,\infty} \\ &\leq \|\mathcal{A}^\top\|_{b,\infty} \cdot \|\mathbf{I}_{NL} - \mathcal{M}(\mathbf{R}_x + \eta \mathcal{Q})\|_{b,\infty}. \end{aligned} \quad (46)$$

We have  $\|\mathcal{A}^\top\|_{b,\infty} = 1$  because  $\mathcal{A}$  is a block left-stochastic matrix. This yields:

$$\begin{aligned} & \rho(\mathcal{A}^\top [I_{NL} - \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q})]) \\ & \leq \|I_{NL} - \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q})\|_{b,\infty} \\ & = \|I_{NL} - \mathcal{M}(\mathcal{R}_x + \eta(I_{NL} - \mathcal{P}))\|_{b,\infty} \\ & \leq \|I_{NL} - \mathcal{M}\mathcal{R}_x - \eta\mathcal{M}\|_{b,\infty} + \eta\|\mathcal{M}\mathcal{P}\|_{b,\infty}. \end{aligned} \quad (47)$$

Consider the first term on the RHS of (47). Since the matrices  $\mathcal{M}$  and  $\mathcal{R}_x$  are block diagonal, it holds from the properties of the block maximum norm [16]:

$$\begin{aligned} & \|I_{NL} - \mathcal{M}\mathcal{R}_x - \eta\mathcal{M}\|_{b,\infty} \\ & = \max_{1 \leq k \leq N} \rho((1 - \eta\bar{\mu}_k)I_L - \bar{\mu}_k \mathcal{R}_{x,k}) \\ & = \max_{1 \leq k \leq N} \max_{1 \leq \ell \leq L} |(1 - \eta\bar{\mu}_k) - \bar{\mu}_k \lambda_\ell(\mathcal{R}_{x,k})| \end{aligned} \quad (48)$$

where  $\bar{\mu}_k \triangleq \mathbb{E}\{\mu_k(i)\}$ , and  $\lambda_\ell(\cdot)$  denotes the  $\ell$ -th eigenvalue of its matrix argument. Consider now the second term on the RHS of (47). Using the submultiplicative property of the block maximum norm, and the fact that  $\mathcal{P}$  is a block right-stochastic matrix, we get

$$\eta\|\mathcal{M}\mathcal{P}\|_{b,\infty} \leq \eta\|\mathcal{M}\|_{b,\infty}. \quad (49)$$

Because  $\mathcal{M}$  is a block diagonal matrix, we further have that

$$\|\mathcal{M}\|_{b,\infty} = \max_{1 \leq k \leq N} \bar{\mu}_k. \quad (50)$$

Combining (48) and (50) we conclude that the algorithm is stable in the mean if

$$\max_{1 \leq k \leq N} \max_{1 \leq \ell \leq L} |1 - \eta\bar{\mu}_k - \bar{\mu}_k \lambda_\ell(\mathcal{R}_{x,k})| + \eta \max_{1 \leq k \leq N} \bar{\mu}_k < 1. \quad (51)$$

In order to simplify this condition, assume that  $\bar{\mu}_k = \bar{\mu}$  for all  $k$ . Condition (51) then reduces to (45). Note that the randomness in the topology does not affect the condition for stability in the mean of the algorithm. ■

### B. Mean-square error behavior analysis

To perform mean-square error analysis over asynchronous networks, compared to synchronous networks [27], new operators with additional properties must be introduced. We shall use the block Kronecker product operator  $\otimes_b$  instead of the Kronecker product  $\otimes$ , and the block vectorization operator  $\text{bvec}(\cdot)$  instead of the vectorization operator  $\text{vec}(\cdot)$ . This is because, as explained in [3], [43], these block operators preserve the locality of the blocks in the original matrix arguments. Recall that if  $\mathbf{X}$  is an  $N \times N$  block matrix with blocks of size  $L \times L$ ,  $\text{bvec}(\mathbf{X})$  vectorizes each block of  $\mathbf{X}$  and stacks the vectors on top of each other. Before proceeding, we recall some properties of these block operators [3], [49]:

For any two  $N \times 1$  block vectors  $\{\mathbf{x}, \mathbf{y}\}$  with blocks of size  $L \times 1$ , we have:

$$\text{bvec}(\mathbf{x}\mathbf{y}^\top) = \mathbf{y} \otimes_b \mathbf{x}. \quad (52)$$

For any  $N \times N$  block-matrices  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$  with blocks of size  $L \times L$ , we have:

$$(\mathbf{A} + \mathbf{B}) \otimes_b (\mathbf{C} + \mathbf{D}) = \mathbf{A} \otimes_b \mathbf{C} + \mathbf{A} \otimes_b \mathbf{D} + \mathbf{B} \otimes_b \mathbf{C} + \mathbf{B} \otimes_b \mathbf{D} \quad (53)$$

$$(\mathbf{A}\mathbf{C}) \otimes_b (\mathbf{B}\mathbf{D}) = (\mathbf{A} \otimes_b \mathbf{B})(\mathbf{C} \otimes_b \mathbf{D}) \quad (54)$$

$$(\mathbf{A} \otimes \mathbf{B}) \otimes_b (\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A} \otimes \mathbf{C}) \otimes (\mathbf{B} \otimes \mathbf{D}) \quad (55)$$

$$\text{trace}(\mathbf{A}\mathbf{B}) = [\text{bvec}(\mathbf{B}^\top)]^\top \text{bvec}(\mathbf{A}) \quad (56)$$

$$\text{bvec}(\mathbf{A}\mathbf{B}\mathbf{C}) = (\mathbf{C}^\top \otimes_b \mathbf{A}) \text{bvec}(\mathbf{B}) \quad (57)$$

$$(\mathbf{A} \otimes_b \mathbf{B})^\top = (\mathbf{A}^\top \otimes_b \mathbf{B}^\top). \quad (58)$$

We now use these properties to evaluate the expectation of some block Kronecker matrix products that will be useful in the sequel:

$$\begin{aligned} \mathcal{M}_I & \triangleq \mathbb{E}\{\mathcal{M}(i) \otimes_b \mathcal{M}(i)\} \\ & = \mathbb{E}\{(\mathcal{M}(i) \otimes I_L) \otimes_b (\mathcal{M}(i) \otimes I_L)\} \\ & \stackrel{(56)}{=} \mathbb{E}\{(\mathcal{M}(i) \otimes \mathcal{M}(i)) \otimes (I_L \otimes I_L)\} \\ & \stackrel{(10)}{=} (\overline{\mathcal{M}} \otimes \overline{\mathcal{M}} + \mathbf{C}_M) \otimes I_{L^2}. \end{aligned} \quad (59)$$

In the same way, we get the following expectations:

$$\mathcal{A}_I \triangleq \mathbb{E}\{\mathcal{A}(i) \otimes_b \mathcal{A}(i)\} = (\overline{\mathcal{A}} \otimes \overline{\mathcal{A}} + \mathbf{C}_A) \otimes I_{L^2}, \quad (60)$$

$$\mathcal{P}_I \triangleq \mathbb{E}\{\mathcal{P}(i) \otimes_b \mathcal{P}(i)\} = (\overline{\mathcal{P}} \otimes \overline{\mathcal{P}} + \mathbf{C}_P) \otimes I_{L^2}. \quad (61)$$

Since  $\mathcal{Q}(i) = I_{NL} - \mathcal{P}(i)$ , we also obtain:

$$\begin{aligned} \mathcal{Q}_I & \triangleq \mathbb{E}\{\mathcal{Q}(i) \otimes_b \mathcal{Q}(i)\} \\ & = (I_{N^2} - I_N \otimes \overline{\mathcal{P}} - \overline{\mathcal{P}} \otimes I_N + \overline{\mathcal{P}} \otimes \overline{\mathcal{P}} + \mathbf{C}_P) \otimes I_{L^2}. \end{aligned} \quad (62)$$

Before concluding these preliminary calculations, let us make some remarks on the stochasticity of matrices considered in the sequel. At each time instant  $i$ , the matrix  $\mathcal{P}(i) \otimes \mathcal{P}(i)$  has nonnegative entries since  $\mathcal{P}(i)$  has nonnegative entries. It follows that  $\mathbb{E}\{\mathcal{P}(i) \otimes \mathcal{P}(i)\} = \overline{\mathcal{P}} \otimes \overline{\mathcal{P}} + \mathbf{C}_P$  has also nonnegative entries, and is right-stochastic since

$$\begin{aligned} (\overline{\mathcal{P}} \otimes \overline{\mathcal{P}} + \mathbf{C}_P) \mathbf{1}_{N^2} & = \mathbb{E}\{(\mathcal{P}(i) \otimes \mathcal{P}(i))(\mathbf{1}_N \otimes \mathbf{1}_N)\} \\ & = \mathbb{E}\{(\mathcal{P}(i) \mathbf{1}_N) \otimes (\mathcal{P}(i) \mathbf{1}_N)\} = \mathbf{1}_{N^2} \end{aligned} \quad (63)$$

In the same token, the matrix  $\overline{\mathcal{A}} \otimes \overline{\mathcal{A}} + \mathbf{C}_A$  is left-stochastic.

To analyze the convergence in mean-square-error sense of the multitask diffusion LMS algorithm (9) over asynchronous networks, we consider the variance of the weight error vector  $\tilde{\mathbf{w}}(i)$  weighted by any positive semi-definite matrix  $\Sigma$ , that is,  $\mathbb{E}\{\|\tilde{\mathbf{w}}(i)\|_\Sigma^2\}$ , where  $\|\tilde{\mathbf{w}}(i)\|_\Sigma^2 \triangleq \tilde{\mathbf{w}}^\top(i) \Sigma \tilde{\mathbf{w}}(i)$ . The freedom in selecting  $\Sigma$  will allow us to extract various types of information about the network and the nodes. By Assumption 1 and using (34), we get:

$$\begin{aligned} \mathbb{E}\{\|\tilde{\mathbf{w}}(i+1)\|_\Sigma^2\} & = \mathbb{E}\{\|\tilde{\mathbf{w}}(i)\|_{\Sigma'}^2\} + \mathbb{E}\{\|g(i)\|_\Sigma^2\} + \\ & \quad \eta^2 \mathbb{E}\{\|\mathbf{r}(i)\|_\Sigma^2\} + 2\eta \mathbb{E}\{\mathbf{r}^\top(i) \Sigma \mathcal{B}(i) \tilde{\mathbf{w}}(i)\} \end{aligned} \quad (64)$$

where  $\Sigma' = \mathbb{E}\{\mathcal{B}^\top(i) \Sigma \mathcal{B}(i)\}$ . Let  $\sigma$  denotes the  $(NL)^2 \times 1$  vector representation of  $\Sigma$  that is obtained by the block

vectorization operator, namely,  $\sigma \triangleq \text{bvec}(\Sigma)$ . In the sequel, it will be more convenient to work with  $\sigma$  than with  $\Sigma$  itself. Let  $\sigma' \triangleq \text{bvec}(\Sigma')$ . Using property (57), we can verify that

$$\sigma' = \mathcal{F}^\top \sigma \quad (65)$$

where  $\mathcal{F}$  is the  $(NL)^2 \times (NL)^2$  matrix given by:

$$\begin{aligned} \mathcal{F} &\triangleq \mathbb{E}\{\mathcal{B}(i) \otimes_b \mathcal{B}(i)\} \\ &\stackrel{(54)}{=} \mathbb{E}\{\mathcal{A}^\top(i) \otimes_b \mathcal{A}^\top(i)\} \mathbb{E}\{[I_{NL} - \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))] \\ &\quad \otimes_b [I_{NL} - \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))]\} \\ &\stackrel{(60), (53)}{=} \mathcal{A}_1^\top [I_{(NL)^2} - I_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) - \\ &\quad \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) \otimes_b I_{NL} + \\ &\quad \mathbb{E}\{\mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))\}] \end{aligned} \quad (66)$$

where using property (54) and the definition of  $\mathcal{M}_1$  in (59), we have

$$\begin{aligned} &\mathbb{E}\{\mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))\} \\ &= \mathcal{M}_1 \mathbb{E}\{(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b (\mathcal{R}_x(i) + \eta \mathcal{Q}(i))\}. \end{aligned} \quad (67)$$

The term on the RHS of equation (67) is proportional to  $\mathcal{M}_1 = \mathbb{E}\{\mathcal{M}(i) \otimes \mathcal{M}(i)\} \otimes I_{L^2}$ , where  $\mathbb{E}\{\mathcal{M}(i) \otimes \mathcal{M}(i)\}$  is an  $N \times N$  block diagonal matrix whose  $k$ -th block is an  $N \times N$  diagonal matrix with  $\ell$ -th entry given by  $\mathbb{E}\{\mu_k(i)\mu_\ell(i)\}$ . It is sufficient for the exposition in this work to focus on the case of sufficiently small step-sizes where terms involving higher order moments of the step-sizes can be ignored. Such approximations are common when analyzing diffusion strategies in the mean-square-error sense (see [16, Section 6.5]). Accordingly, the last term in (66) can be neglected and we continue our discussion by letting

$$\mathcal{F} \approx \mathcal{A}_1^\top [I_{(NL)^2} - I_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) - \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) \otimes_b I_{NL}]. \quad (68)$$

Consider next the second term on the RHS of (64). We can write:

$$\mathbb{E}\{\|g(i)\|_\Sigma^2\} = \text{trace}\{\Sigma \mathbb{E}\{g(i) g^\top(i)\}\} \stackrel{(56)}{=} g_b^\top \sigma \quad (69)$$

where  $g_b = \text{bvec}(\mathbb{E}\{g(i) g^\top(i)\})$ . Using expression (32) and the definitions of  $\mathcal{M}_1$  and  $\mathcal{A}_1$  in (59) and (60), we have

$$\begin{aligned} g_b &= \text{bvec}(\mathbb{E}\{(\mathcal{A}^\top(i) \mathcal{M}(i) p_{xz}(i) p_{xz}^\top(i) \mathcal{M}(i) \mathcal{A}(i))\} \\ &\stackrel{(57)}{=} \mathbb{E}\{(\mathcal{A}^\top(i) \otimes_b \mathcal{A}^\top(i)) \text{bvec}(\mathcal{M}(i) p_{xz}(i) p_{xz}^\top(i) \mathcal{M}(i))\} \\ &\stackrel{(57), (58)}{=} \mathcal{A}_1^\top \mathbb{E}\{(\mathcal{M}(i) \otimes_b \mathcal{M}(i)) \text{bvec}(p_{xz}(i) p_{xz}^\top(i))\} \\ &= \mathcal{A}_1^\top \mathcal{M}_1 \text{bvec}(\mathcal{S}), \end{aligned} \quad (70)$$

where  $\mathcal{S} \triangleq \mathbb{E}\{p_{xz}(i) p_{xz}^\top(i)\} = \text{diag}\{\sigma_{z,k}^2 \mathbf{R}_{x,k}\}_{k=1}^N$ . Let us examine now the third term on the RHS of (64):

$$\mathbb{E}\{\|r(i)\|_\Sigma^2\} = \text{trace}\{\Sigma \mathbb{E}\{r(i) r^\top(i)\}\} \stackrel{(56)}{=} r_b^\top \sigma \quad (71)$$

where  $r_b = \text{bvec}(\mathbb{E}\{r(i) r^\top(i)\})$ . Using expression (33), property (57), and the definitions of  $\mathcal{M}_1$ ,  $\mathcal{A}_1$ , and  $\mathcal{Q}_1$  in (59), (60), and (62), and proceeding as in (70), we obtain the following expression:

$$r_b = \mathcal{A}_1^\top \mathcal{M}_1 \mathcal{Q}_1 \text{bvec}(w^*(w^*)^\top). \quad (72)$$

Consider now the fourth term  $\mathbb{E}\{r^\top(i) \Sigma \mathcal{B}(i) \tilde{w}(i)\}$ . We have:

$$\begin{aligned} \mathbb{E}\{r^\top(i) \Sigma \mathcal{B}(i) \tilde{w}(i)\} &= \mathbb{E}\{\text{bvec}(r^\top(i) \Sigma \mathcal{B}(i) \tilde{w}(i))\} \\ &\stackrel{(57)}{=} \mathbb{E}\{(\mathcal{B}(i) \tilde{w}(i))^\top \otimes_b r^\top(i)\} \sigma \\ &\stackrel{(58)}{=} \mathbb{E}\{\mathcal{B}(i) \tilde{w}(i) \otimes_b r(i)\}^\top \sigma \\ &\stackrel{(54)}{=} \mathbb{E}\{\tilde{w}(i) \otimes_b 1\}^\top \mathbb{E}\{\mathcal{B}(i) \otimes_b r(i)\}^\top \sigma \\ &= \mathbb{E}\{\tilde{w}(i)\}^\top \mathbb{E}\{\mathcal{B}(i) \otimes_b r(i)\}^\top \sigma \end{aligned} \quad (73)$$

with

$$\begin{aligned} &\mathbb{E}\{\mathcal{B}(i) \otimes_b r(i)\} \\ &= \mathbb{E}\{\mathcal{A}^\top(i) [I_{NL} - \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))] \otimes_b \\ &\quad \mathcal{A}^\top(i) \mathcal{M}(i) \mathcal{Q}(i) w^*\} \\ &\stackrel{(54)}{=} \mathcal{A}_1^\top \mathbb{E}\{[I_{NL} - \mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i))] \otimes_b \\ &\quad \mathcal{M}(i) \mathcal{Q}(i) w^*\} \\ &\stackrel{(53)}{=} \mathcal{A}_1^\top [(I_{NL} \otimes_b \mathcal{M} \mathcal{Q} w^*) - \\ &\quad \mathbb{E}\{\mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b \mathcal{M}(i) \mathcal{Q}(i) w^*\}], \end{aligned} \quad (74)$$

where

$$\begin{aligned} &\mathbb{E}\{\mathcal{M}(i)(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b \mathcal{M}(i) \mathcal{Q}(i) w^*\} \\ &\stackrel{(54)}{=} \mathcal{M}_1 \mathbb{E}\{(\mathcal{R}_x(i) + \eta \mathcal{Q}(i)) \otimes_b \mathcal{Q}(i) w^*\} \\ &\stackrel{(53)}{=} \mathcal{M}_1 ((\mathcal{R}_x \otimes_b \mathcal{Q} w^*) + \eta \mathbb{E}\{\mathcal{Q}(i) \otimes_b \mathcal{Q}(i) w^*\}) \\ &\stackrel{(54)}{=} \mathcal{M}_1 ((\mathcal{R}_x \otimes_b \mathcal{Q} w^*) + \eta \mathcal{Q}_1 (I_{NL} \otimes_b w^*)). \end{aligned} \quad (75)$$

Finally, combining (74) and (75) and introducing the notation  $\mathcal{K}$ , we get

$$\begin{aligned} \mathcal{K} &\triangleq \mathbb{E}\{\mathcal{B}(i) \otimes_b r(i)\} \\ &= \mathcal{A}_1^\top [(I_{NL} \otimes_b \mathcal{M} \mathcal{Q} w^*) - \mathcal{M}_1 ((\mathcal{R}_x \otimes_b \mathcal{Q} w^*) + \\ &\quad \eta \mathcal{Q}_1 (I_{NL} \otimes_b w^*))]. \end{aligned} \quad (76)$$

Relation (64) can be written in a more compact form as

$$\mathbb{E}\{\|\tilde{w}(i+1)\|_\Sigma^2\} = \mathbb{E}\{\|\tilde{w}(i)\|_{\mathcal{F}^\top \sigma}^2\} + y^\top(i) \sigma, \quad (77)$$

where  $y(i)$  is the  $(LN)^2 \times 1$  vector given by:

$$y(i) \triangleq g_b + \eta^2 r_b + 2\eta \mathcal{K} \mathbb{E}\{\tilde{w}(i)\}. \quad (78)$$

In the sequel, we shall use the notations  $\|\cdot\|_\Sigma$  and  $\|\cdot\|_\sigma$  interchangeably.

**Theorem 2. (Mean-square stability)** Assume data model (1) and Assumption 1 hold. Assume further that the upper bounds on the step-sizes,  $\{\mu_{\max,k}\}$ , are sufficiently small such that approximation (68) is justified by ignoring higher-order powers of the step-sizes, and (77) can be used as a reasonable representation for the dynamics of the weighted mean-square error. Then, the asynchronous diffusion multitask algorithm (9) is mean-square stable if the matrix  $\mathcal{F}$  defined by (68) is stable.

*Proof:* Provided that  $\mathcal{F}$  is stable, recursion (77) is stable if  $y^\top(i) \sigma$  is bounded. Since  $\eta, g_b, r_b, \mathcal{K}$ , and  $\sigma$  are finite and constant terms, the boundedness of  $y^\top(i) \sigma$  depends on  $\mathbb{E}\{\tilde{w}(i)\}$  being bounded. We know from (35) that  $\mathbb{E}\{\tilde{w}(i)\}$  is uniformly bounded because (35) is a Bounded Input Bounded Output (BIBO) stable recursion with a bounded driving term



$\eta \mathcal{A}^\top \mathcal{M} \mathcal{Q} \mathbf{w}^*$ . It follows that  $\mathbf{y}^\top(i) \boldsymbol{\sigma}$  is uniformly bounded. As a result,  $\mathbb{E}\{\|\tilde{\mathbf{w}}(i+1)\|_{\boldsymbol{\sigma}}^2\}$  converges to a bounded value as  $i \rightarrow \infty$ , and the algorithm is mean-square stable.

The stability of  $\mathcal{F}$  is studied in Appendix B. It is worth noting that, due to the Kronecker covariance matrix  $\mathbf{C}_A$ , the matrix  $\mathcal{F}$  cannot be approximated by  $\mathbf{B} \otimes \mathbf{B}$  as in the synchronous case [16], [27]. Moreover, deriving a condition that ensures the stability of  $\mathcal{F}$  in a multitask setting is more challenging than in the single-task setting [43] due to the presence of the non-block diagonal matrix  $\mathcal{Q}$  in the second term on the RHS of (68). ■

**Theorem 3. (Transient network performance)** Consider sufficiently small step-sizes that ensure mean and mean-square stability. The variance curve defined by  $\zeta(i) = \mathbb{E}\{\|\tilde{\mathbf{w}}(i+1)\|_{\boldsymbol{\sigma}}^2\}$  evolves according to the following recursion for  $i \geq 0$ :

$$\begin{aligned} \zeta(i+1) &= \zeta(i) + \|\tilde{\mathbf{w}}(0)\|_{(\mathcal{F}^\top - \mathbf{I}_{(NL)^2})(\mathcal{F}^\top)^\top \boldsymbol{\sigma}}^2 + (\mathbf{y}^\top(i) + \boldsymbol{\Gamma}(i)) \boldsymbol{\sigma} \\ &\quad (79) \end{aligned}$$

where  $\boldsymbol{\Gamma}(i+1)$  is updated as follows:

$$\boldsymbol{\Gamma}(i+1) = \boldsymbol{\Gamma}(i) \mathcal{F}^\top + \mathbf{y}^\top(i) (\mathcal{F}^\top - \mathbf{I}_{(NL)^2}), \quad (80)$$

with the initial conditions  $\zeta(0) = \|\tilde{\mathbf{w}}(0)\|_{\boldsymbol{\sigma}}^2$  and  $\boldsymbol{\Gamma}(0) = \mathbf{0}_{(NL)^2}^\top$ . The network mean-square deviation (MSD) is obtained by setting  $\boldsymbol{\sigma} = \text{bvec}(\boldsymbol{\Sigma})$  with  $\boldsymbol{\Sigma} = \frac{1}{N} \mathbf{I}_{NL}$ .

*Proof:* The argument is similar to the proof of Theorem 3 in [27]. ■

**Theorem 4. (Steady-state network performance)** Assume sufficiently small step-sizes to ensure mean and mean-square convergence. Then, the steady-state performance for multitask diffusion LMS (9) applied to asynchronous network is given by:

$$\zeta^* = (\mathbf{g}_b + \eta^2 \mathbf{r}_b + 2\eta \mathcal{K} \mathbb{E}\{\tilde{\mathbf{w}}(\infty)\})^\top (\mathbf{I}_{(NL)^2} - \mathcal{F}^\top)^{-1} \boldsymbol{\sigma}. \quad (81)$$

where  $\mathbb{E}\{\tilde{\mathbf{w}}(\infty)\}$  is given by (44). The network mean-square deviation (MSD) is obtained by setting  $\boldsymbol{\sigma} = \text{bvec}(\boldsymbol{\Sigma})$  with  $\boldsymbol{\Sigma} = \frac{1}{N} \mathbf{I}_{NL}$ .

*Proof:* The steady-state network performance with metric  $\boldsymbol{\sigma}$  is defined as:

$$\zeta^* = \lim_{i \rightarrow \infty} \mathbb{E}\{\|\tilde{\mathbf{w}}(i)\|_{\boldsymbol{\sigma}}^2\}. \quad (82)$$

From the recursive expression (77), we obtain as  $i \rightarrow \infty$ :

$$\begin{aligned} \lim_{i \rightarrow \infty} \mathbb{E}\{\|\tilde{\mathbf{w}}(i)\|_{(\mathbf{I}_{(NL)^2} - \mathcal{F}^\top) \boldsymbol{\sigma}}^2\} \\ = (\mathbf{g}_b + \eta^2 \mathbf{r}_b + 2\eta \mathcal{K} \mathbb{E}\{\tilde{\mathbf{w}}(\infty)\})^\top \boldsymbol{\sigma}. \end{aligned} \quad (83)$$

To obtain (82), we replace  $\boldsymbol{\sigma}$  in (83) by  $(\mathbf{I}_{(NL)^2} - \mathcal{F}^\top)^{-1} \boldsymbol{\sigma}$ . ■

Before moving on to the presentation of experimental results, note that the performance of the *synchronous* multitask algorithm over the *mean-graph* topology can be obtained by setting  $\mathbf{C}_A$ ,  $\mathbf{C}_M$ , and  $\mathbf{C}_P$  to zero in (59)–(61).

## IV. SIMULATION RESULTS

### A. Illustrative example

We adopt the same clustered multitask network as [27] in our simulations. As shown in Figure 2, the network consists of 10 nodes divided into 4 clusters:  $\mathcal{C}_1 = \{1, 2, 3\}$ ,  $\mathcal{C}_2 = \{4, 5, 6\}$ ,  $\mathcal{C}_3 = \{7, 8\}$ ,  $\mathcal{C}_4 = \{9, 10\}$ . The unknown parameter vector  $\mathbf{w}_{\mathcal{C}_i}^*$  of each cluster is of size  $2 \times 1$ , and has the following form:  $\mathbf{w}_{\mathcal{C}_i}^* = \mathbf{w}_0 + \delta \mathbf{w}_{\mathcal{C}_i}$  with  $\mathbf{w}_0 = [0.5, -0.4]^\top$ ,  $\delta \mathbf{w}_{\mathcal{C}_1} = [0.0287, -0.005]^\top$ ,  $\delta \mathbf{w}_{\mathcal{C}_2} = [0.0234, 0.005]^\top$ ,  $\delta \mathbf{w}_{\mathcal{C}_3} = [-0.0335, 0.0029]^\top$ , and  $\delta \mathbf{w}_{\mathcal{C}_4} = [0.0224, 0.00347]^\top$ . The input and output data at each node  $k$  are related via the linear regression model:  $d_k(i) = \mathbf{x}_k^\top(i) \mathbf{w}_k^* + z_k(i)$  where  $\mathbf{w}_k^* = \mathbf{w}_{\mathcal{C}(k)}^*$ . The regressors are zero-mean  $2 \times 1$  random vectors governed by a Gaussian distribution with covariance matrices  $\mathbf{R}_{x,k} = \sigma_{x,k}^2 \mathbf{I}_L$ . The variances  $\sigma_{x,k}^2$  are shown in Figure 2. The background noises  $z_k(i)$  are independent and identically distributed zero-mean Gaussian random variables, independent of any other signals. The corresponding variances are given in Figure 2.

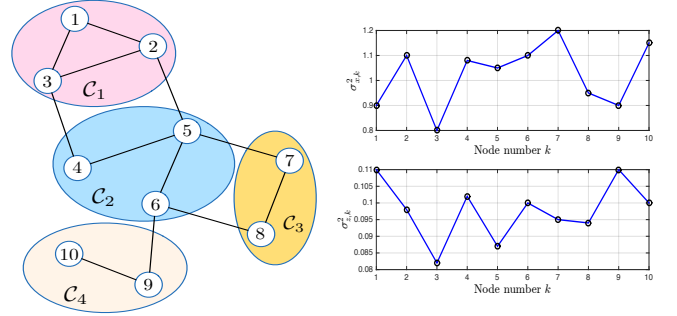


Fig. 2. Experimental setup. Left: Network topology. Right: Regression and noise variances.

We considered the Bernoulli asynchronous model described in Appendix A. We set the coefficient  $a_{\ell k}$  in (90) such that  $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$  for all  $\ell \in (\mathcal{N}_k \cap \mathcal{C}(k))$ , where  $|\mathcal{N}_k \cap \mathcal{C}(k)|$  denotes the cardinality of the set  $\mathcal{N}_k \cap \mathcal{C}(k)$ . Then we set the regularization factors  $\rho_{k\ell}$  in (94) as follows. If  $\mathcal{N}_k \setminus \mathcal{C}(k) \neq \emptyset$ ,  $\rho_{k\ell}$  was set to  $\rho_{k\ell} = |\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$  for all  $\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)$ , and to  $\rho_{k\ell} = 0$  for any other  $\ell$ . If  $\mathcal{N}_k \setminus \mathcal{C}(k) = \emptyset$ , these factors were set to  $\rho_{kk} = 1$  and to  $\rho_{k\ell} = 0$  for all  $\ell \neq k$ . This usually leads to asymmetrical regularization factors. The parameters of the Bernoulli distribution governing the step-sizes  $\mu_k(i)$  were the same over the network, that is, we set  $\mu_k$  in (88) to 0.03 for all  $k$ . The regularization strength  $\eta$  was set to 1. The MSD learning curves were averaged over 100 Monte-Carlo runs. The transient MSD curves were obtained with Theorem 3, and the steady-state MSD was estimated with Theorem 4. In Figure 3 (left), we report the network MSD learning curves for 3 different cases:

- Case 1: 50% idle:  $q_k = p_{\ell k} = r_{k\ell} = 0.5$ ;
- Case 2: 30% idle:  $q_k = p_{\ell k} = r_{k\ell} = 0.7$ ;
- Case 3: no idle nodes:  $q_k = p_{\ell k} = r_{k\ell} = 1$ .

We observe that the simulation results match well the theoretical results. Furthermore, the performance of the network is influenced by the probability of occurrence of random events. In Figure 3 (right), the asynchronous algorithm in Case 2 is



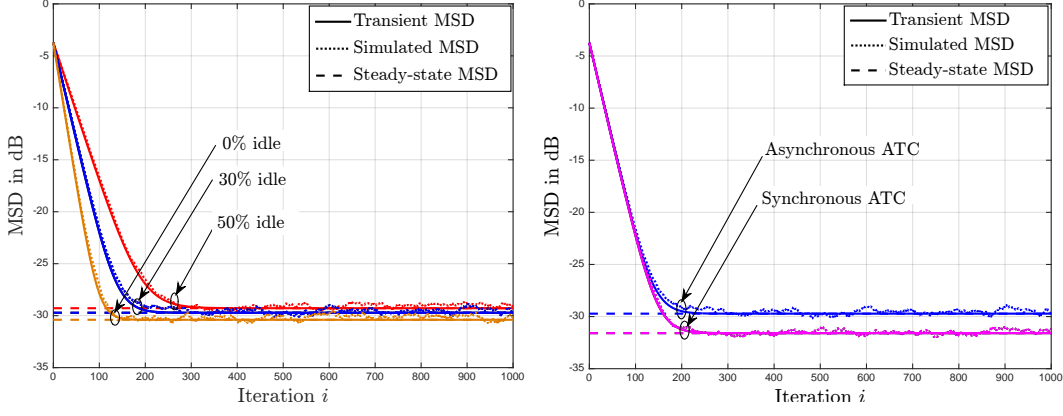


Fig. 3. Left: Comparison of asynchronous network MSD under 50% idle, 30% idle, and 0% idle. Right: Network MSD comparison of asynchronous network under 30% idle and the corresponding synchronous network.

compared with its synchronous version obtained from (7) by setting  $\mu_k$ ,  $a_{\ell k}$ , and  $\rho_{k\ell}$  to the expected values  $\bar{\mu}_k = \mathbb{E}\{\mu_k(i)\}$ ,  $\bar{a}_{\ell k} = \mathbb{E}\{a_{\ell k}(i)\}$ , and  $\bar{\rho}_{k\ell} = \mathbb{E}\{\rho_{k\ell}(i)\}$ , respectively. Although both algorithms show the same convergence rate, the asynchronous algorithm suffers from degradation in its MSD performance caused by the additional randomness throughout the adaptation process.

### B. Multitask learning benefit

In this section we provide an example to show the benefit of multitask learning. We consider a network consisting of  $N = 100$  nodes grouped into  $Q = 3$  clusters such that  $\mathcal{C}_1 = \{1, \dots, 70\}$ ,  $\mathcal{C}_2 = \{71, \dots, 90\}$ , and  $\mathcal{C}_3 = \{91, \dots, 100\}$ . The physical connections are defined by the connectivity matrix represented in Figure 4. The inputs  $\mathbf{x}_k(i)$  were zero-mean  $21 \times 1$  random vectors governed by a Gaussian distribution with covariance matrix  $\mathbf{R}_{\mathbf{x},k} = \sigma_{\mathbf{x},k}^2 \mathbf{I}_{21}$ , where  $\sigma_{\mathbf{x},k}^2$  were randomly chosen in the interval  $[1, 1.4]$ . The noises  $z_k(i)$  were i.i.d. zero-mean Gaussian random variables, independent of any other signal with variances  $\sigma_{z,k}^2$  randomly chosen in the interval  $[0.1, 0.15]$ . The  $21 \times 1$  unknown parameter vectors were chosen as:  $\mathbf{w}_{\mathcal{C}_1}^* = \mathbf{w}_0 = [\mathbf{1}_{1 \times 3}, \mathbf{0}_{1 \times 3}, 2 \cdot \mathbf{1}_{1 \times 3}, \mathbf{0}_{1 \times 3}, -\mathbf{1}_{1 \times 3}, \mathbf{0}_{1 \times 3}, -2 \cdot \mathbf{1}_{1 \times 3}]^T$ ,  $\mathbf{w}_{\mathcal{C}_2}^* = \mathbf{w}_0 + \delta \mathbf{w}$ ,  $\mathbf{w}_{\mathcal{C}_3}^* = \mathbf{w}_0 - \delta \mathbf{w}$  where  $\delta \mathbf{w}$  was randomly generated such that  $\|\delta \mathbf{w}\|_\infty = \max_i |\delta \mathbf{w}_i| = 0.03$ .

We considered the Bernoulli asynchronous model. The coefficients  $\{a_{\ell k}\}$  and  $\{\rho_{k\ell}\}$  in (90) and (94), respectively, were generated in the same manner as in IV-A. Parameters  $\mu_k$  and  $q_k$  in (88) were set to  $\mu_k = 1/30$ ,  $q_k = 0.8$  for nodes in the first cluster,  $\mu_k = 2/45$ ,  $q_k = 0.6$  for nodes in the second cluster, and  $\mu_k = 1/15$ ,  $q_k = 0.4$  for nodes in the third cluster. The probabilities  $\{p_{\ell k}\}$  in (90) were  $p_{\ell k} = 0.8$  for links in the first cluster,  $p_{\ell k} = 0.6$  for links in the second cluster, and  $p_{\ell k} = 0.4$  for links in the third cluster. The probability that a link connecting two nodes belonging to neighboring clusters drops was  $1 - r_{k\ell} = 0.25$ . The simulated curves were obtained by averaging over 150 Monte-Carlo runs.

In Figure 5 (left), we compare two algorithms: the asynchronous diffusion strategy without regularization (obtained from (9) by setting  $\eta = 0$ ) and its synchronous counterpart

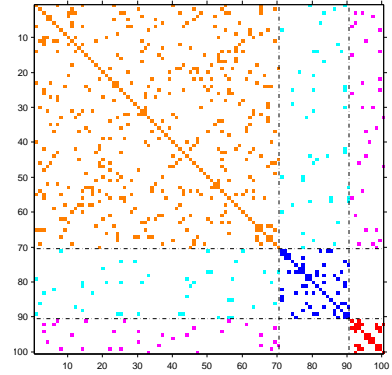


Fig. 4. Connectivity matrix of the network. The orange, blue, and red elements correspond to links within  $\mathcal{C}_1$ ,  $\mathcal{C}_2$ , and  $\mathcal{C}_3$ , respectively. The cyan elements correspond to links between  $\mathcal{C}_1$  and  $\mathcal{C}_2$  and the magenta elements correspond to links between  $\mathcal{C}_1$  and  $\mathcal{C}_3$ . No links between  $\mathcal{C}_2$  and  $\mathcal{C}_3$ .

(obtained from (9) by setting  $\eta = 0$  and replacing  $\mu_k(i)$ ,  $a_{\ell k}(i)$  by  $\bar{\mu}_k$ ,  $\bar{a}_{\ell k}$ ). As shown in this figure, the performance is highly deteriorated in the third cluster and slightly deteriorated in the first cluster because  $\mathcal{C}_3$  is more susceptible to random events. In Figure 5 (right), we compare two algorithms: the asynchronous diffusion strategy with regularization (obtained from (9) by setting  $\eta = 2$ ) and the same synchronous algorithm as in the left plot. As shown in this figure, the cooperation between clusters improves the performance of each cluster so that gaps appearing in the left plot are reduced. In other words,  $\mathcal{C}_2$  and  $\mathcal{C}_3$  benefit from the high performance levels achieved by  $\mathcal{C}_1$ . This can be justified by two arguments: a large number of nodes is employed to collectively estimate  $\mathbf{w}_{\mathcal{C}_1}^*$  and the probabilities associated with random events in  $\mathcal{C}_1$  are small. As a conclusion, when tasks between neighboring clusters are similar, cooperation among clusters improves the learning especially for clusters where asynchronous events occur frequently.

### C. Circular arcs localization

In this section, we consider the problem of adaptive surface localization over asynchronous networks. When dealing with a smooth target surface, we can expect that promoting the

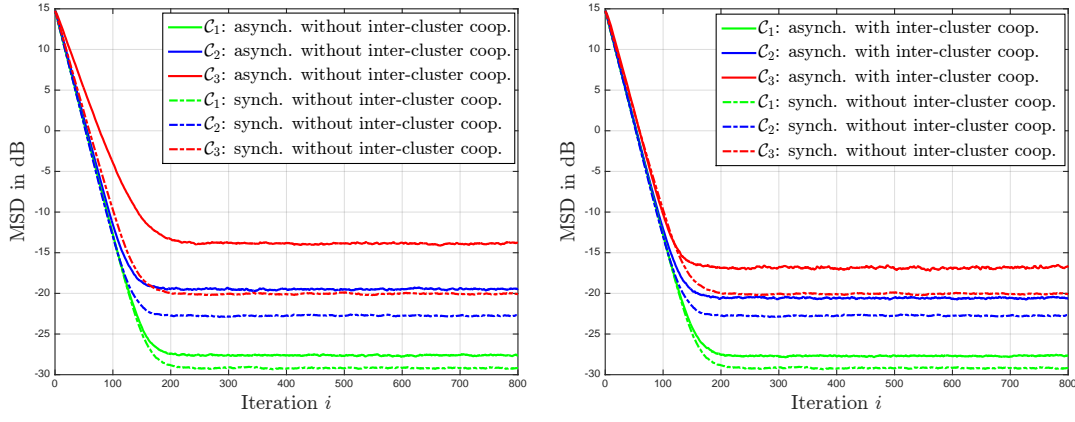


Fig. 5. Cluster learning curves. Left: Comparison of the asynchronous multitask diffusion LMS (9) without inter-cluster cooperation ( $\eta = 0$ ) and its synchronous counterpart. Right: Comparison of the asynchronous multitask diffusion LMS (9) with inter-cluster cooperation ( $\eta \neq 0$ ) and the multitask diffusion LMS (9) without inter-cluster cooperation ( $\eta = 0$ ).

smoothness of the graph signal will improve the performance of the network [27]. In the following, we consider an arc localization application where the radius of the arc is changing over time, and we illustrate the influence of the random events on the learning behavior and tracking ability of the network.

Let us denote by  $\mathcal{L} = [\theta_0, \theta_1]$  an arc of circle with radius  $R$  and subtending an angle  $\theta = \theta_1 - \theta_0$  with the circle center  $\mathbf{w}_o$ . Let us decompose  $\mathcal{L}$  into  $Q$  sub-arcs  $\mathcal{L}_q$  with radius  $R$  and subtending an angle  $\delta \ll \theta$  with  $\mathbf{w}_o$ . In order to estimate the location of  $\mathcal{L}$ , and for sufficiently small  $\delta$ , it is sufficient to estimate the location of each of these  $Q$  sub-arcs by solving a point target localization problem. This can be done by employing a network of  $N$  nodes, composed of  $Q$  clusters, where nodes of each cluster  $\mathcal{C}_q$  are interested in locating  $\mathcal{L}_q$  by estimating a parameter vector  $\mathbf{w}_{\mathcal{C}_q}^*$ . Let us consider node  $k$  belonging to cluster  $\mathcal{C}_q$ . At each time instant  $i$ , node  $k$  gets noisy measurements  $\{d_k(i), \mathbf{u}_k(i)\}$  that are related via the linear data model [16]:

$$d_k(i) = \mathbf{u}_k^\top(i) \mathbf{w}_{\mathcal{C}_q}^* + v_k(i), \quad (84)$$

where  $v_k(i)$  is a zero-mean temporally and spatially independent Gaussian noise with variance  $\sigma_{v,k}^2$ ,  $\mathbf{u}_k(i)$  is a noisy measurement of the unit-norm direction vector of  $\mathbf{u}_k$  pointing from agent  $k$  to the target  $\mathbf{w}_{\mathcal{C}_q}^*$  given by:

$$\mathbf{u}_k(i) = \mathbf{u}_k + \alpha_k(i) \mathbf{u}_k^\perp + \beta_k(i) \mathbf{u}_k, \quad (85)$$

with  $\mathbf{u}_k$  given by  $\mathbf{u}_k = (\mathbf{w}_{\mathcal{C}_q}^* - \mathbf{n}_k) / \|\mathbf{w}_{\mathcal{C}_q}^* - \mathbf{n}_k\|$  where  $\mathbf{n}_k$  is the location vector of node  $k$ ,  $\mathbf{u}_k^\perp$  denoting a unit norm vector that lies in the same space as  $\mathbf{u}_k$  and whose direction is perpendicular to  $\mathbf{u}_k$ . The variables  $\alpha_k(i)$  and  $\beta_k(i)$  are zero-mean independent Gaussian random variables of variances  $\sigma_{\alpha,k}^2$  and  $\sigma_{\beta,k}^2$ , respectively. The amount of perturbation along the parallel direction is assumed to be small compared to the amount of perturbation along the perpendicular direction, that is,  $\sigma_{\beta,k}^2 \ll \sigma_{\alpha,k}^2$ .

To show the effects of randomness at the level of nodes and links, we considered a network of 100 nodes grouped into  $Q = 10$  clusters, located over arcs of radiuses uniformly distributed between  $3R_0$  and  $5R_0$  given  $R_0$ . Angular parameters  $\theta_0$  and

$\theta_1$  were set to  $13\pi/8$  and  $15\pi/8$ , respectively. The network topology is shown in Figure 6. The noise variances were set to  $\sigma_{v,k}^2 = 0.2$ ,  $\sigma_{\alpha,k}^2 = 0.05$ , and  $\sigma_{\beta,k}^2 = 0.005$ , for all  $k$ . We considered a Bernoulli asynchronous model. The coefficients  $a_{\ell k}$  in (90) were set to  $|\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$  for intra-cluster links, and to zero for inter-cluster links. The regularization factors  $\rho_{k\ell}$  in (94) were set to  $|\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$ . The probabilities of success  $q_k$ ,  $p_{\ell k}$ , and  $r_{k\ell}$  were identically set to 0.5.

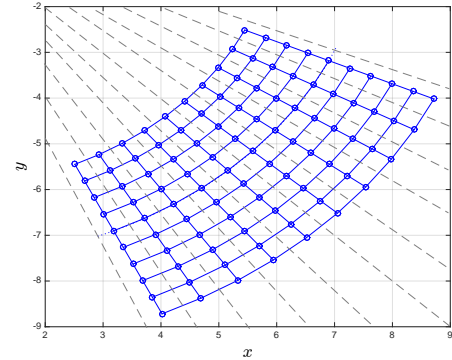


Fig. 6. Network topology consisting of 10 clusters: circles for nodes, solid lines for links, and dashed lines for cluster boundaries.

The MSD learning curves were averaged over 200 Monte-Carlo runs. We ran the synchronous and asynchronous multitask algorithms in two different situations. For the first one, we set the regularization strength  $\eta$  to zero, that is, we did not allow any cooperation between neighboring clusters. In the second one, we set the regularization strength  $\eta$  to 0.5. For comparison purposes, we also ran the noncooperative LMS, which was obtained by setting  $\mathbf{A}(i) = \mathbf{P}(i) = \mathbf{I}_N$  for all  $i$ , and the standard diffusion LMS [17]. In both cases, synchronous and asynchronous algorithms were also considered. Each synchronous algorithm was derived from its asynchronous counterpart by making  $\mu_k(i)$ ,  $a_{\ell k}(i)$ , and  $\rho_{k\ell}(i)$  deterministic quantities equal to  $\bar{\mu}_k$ ,  $\bar{a}_{\ell k}$  and  $\bar{\rho}_{k\ell}$ , respectively. In order to illustrate the tracking ability of the algorithms, we modified the radius  $R$  of  $\mathcal{L}$  every 500 iterations such that:

for  $i \in [0, 500]$ ,  $R = 0.5R_0$ , for  $i \in [500, 1000]$ ,  $R = R_0$ , for  $i \in [1000, 1500]$ ,  $R = 1.5R_0$ , and for  $i \in [1500, 2000]$ ,  $R = 2R_0$ . Note that varying  $R$  has an effect on the level of similarity between neighboring tasks when characterized by  $\|w_{\mathcal{C}_i}^* - w_{\mathcal{C}_j}^*\|^2$ , where  $\mathcal{C}_i$  and  $\mathcal{C}_j$  denote two neighboring clusters. Indeed,  $w_{\mathcal{C}_j}^*$  can be expressed as:

$$w_{\mathcal{C}_j}^* = w_o + R \begin{pmatrix} \cos\left(\theta_0 + \frac{\theta}{Q}\left(j - \frac{1}{2}\right)\right) \\ \sin\left(\theta_0 + \frac{\theta}{Q}\left(j - \frac{1}{2}\right)\right) \end{pmatrix}, \quad \forall j = 1, \dots, Q, \quad (86)$$

where  $\theta = \theta_1 - \theta_0$ . With the topology shown in Fig. 6, we obtain:

$$\|w_{\mathcal{C}_i}^* - w_{\mathcal{C}_j}^*\|^2 = R^2(2 - 2\cos(\theta/Q)). \quad (87)$$

Figure 7 shows that cooperation among clusters improved the network MSD performance and endowed the network with robustness towards asynchronous events. We also observe that the performance of the standard diffusion LMS algorithm deteriorates when the level of similarity between tasks decreases. Figure 8 depicts the estimated arc when  $R = R_0$  for the following algorithms in an asynchronous setting: noncooperative LMS obtained by setting  $A(i) = P(i) = I_N$  for all  $i$ , standard diffusion LMS [17], and multitask diffusion LMS (9). In each case, the results were averaged over 150 Monte-Carlo runs and over 50 samples after convergence. The multitask diffusion algorithm outperformed the non cooperative LMS and the standard diffusion. The standard diffusion was not able to estimate the location of the target since it is a single task algorithm. It is shown in [29] that standard diffusion LMS converges to a Pareto optimal solution when it is applied to multitask problems.

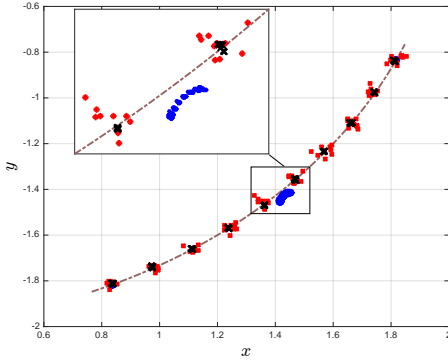


Fig. 8. Target estimation results ( $R = R_0 = 2$ ) over asynchronous network: black cross sign for multitask diffusion (9), red asterisk sign for non-cooperative, and blue circle sign for standard diffusion [17].

Finally, in order to show the effects of the number of clusters (or tasks) on the performance of the network, we considered 2 additional experimental setups. In the first one represented in Figure 9 (left), the number of tasks was set to 5, that is, the arc  $\mathcal{L}$  was decomposed into 5 sub-arcs. In the second one depicted in Figure 9 (right), the number of clusters was set to 15. Except for these changes, we considered the same experimental setup as before. Every 500 time steps, the radius  $R$  of the arc

was modified as before in order to decrease the similarity level between tasks. The learning curves of the algorithms considered in Figure 7 are reported in Figure 10. As expected, it can be observed that the larger the number of clusters is, the more efficient the collaboration between clusters becomes. The benefits of inter-cluster cooperation decreases when the number of clusters becomes small.

## V. CONCLUSION AND PERSPECTIVES

In this paper, we considered multitask problems where networks are able to handle situations beyond the case where the nodes estimate a unique parameter vector over the network. We introduced a general model for asynchronous behavior with random step-sizes, combination coefficients, and co-regularization factors. We then carried out a convergence analysis of the asynchronous multitask algorithm in the mean and mean-square-error sense, and we derived conditions for convergence. Several open problems still have to be solved for specific applications. For instance, it would be interesting to investigate how nodes can autonomously adjust co-regularization factors between neighboring clusters in order to optimize the learning performance. It would also be advantageous to consider alternative co-regularizers in order to promote properties such as sparsity or block sparsity, and to analyze the convergence behavior of the resulting algorithms.

## APPENDIX A THE BERNOULLI MODEL

In this model, the step-sizes  $\{\mu_k(i)\}$  are distributed as follows:

$$\mu_k(i) = \begin{cases} \mu_k, & \text{with probability } q_k \\ 0, & \text{with probability } 1 - q_k \end{cases} \quad (88)$$

where  $\mu_k$  is a fixed value. This probability distribution allows us to model random “on-off” behavior by each agent  $k$  due to power saving strategies or random agent failures. We assume that the step-sizes  $\mu_k(i)$  are spatially uncorrelated for different  $k$ . At each iteration  $i$ , the mean of the step-size  $\mu_k(i)$  is  $\bar{\mu}_k = \mu_k q_k$ , and the covariance between  $\mu_k(i)$  and  $\mu_\ell(i)$  is:

$$\begin{aligned} c_{\mu,k,\ell} &\triangleq \mathbb{E}\{(\mu_k(i) - \bar{\mu}_k)(\mu_\ell(i) - \bar{\mu}_\ell)\} \\ &= \begin{cases} \mu_k^2 q_k (1 - q_k), & \text{if } \ell = k \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (89)$$

Furthermore, combination weights  $\{a_{\ell k}(i)\}$  are distributed as follows:

$$a_{\ell k}(i) = \begin{cases} a_{\ell k}, & \text{with probability } p_{\ell k} \\ 0, & \text{with probability } 1 - p_{\ell k} \end{cases} \quad (90)$$

for any  $\ell \in \mathcal{N}_k^-(i) \cap \mathcal{C}(k)$ , where  $0 < a_{\ell k} < 1$  a fixed coefficient. The coefficients  $\{a_{\ell k}(i)\}$  are spatially uncorrelated for different  $\ell$  and  $k$ . Node  $k$  adjusts its own combination coefficient to ensure that the sum of its neighboring coefficients is equal to one as follows:

$$a_{kk}(i) = 1 - \sum_{\ell \in \mathcal{N}_k^-(i) \cap \mathcal{C}(k)} a_{\ell k}(i) \geq 0. \quad (91)$$

The probability distribution (90) allows us to model a random “on-off” status for links within clusters at time  $i$  due to

communication cost saving strategies or random link failures. With this model, we are giving the opportunity to each agent  $k$  to randomly choose a subset of neighbors that belong to its cluster to perform the combination step. At each iteration  $i$ , the mean of the coefficient  $a_{\ell k}(i)$  is given by:

$$\bar{a}_{\ell k} = \begin{cases} a_{\ell k} p_{\ell k}, & \text{if } \ell \in \mathcal{N}_k^- \cap \mathcal{C}(k) \\ 1 - \sum_{\ell \in \mathcal{N}_k^- \cap \mathcal{C}(k)} a_{\ell k} p_{\ell k}, & \text{if } \ell = k \\ 0, & \text{otherwise.} \end{cases} \quad (92)$$

and the covariance between  $a_{\ell k}(i)$  and  $a_{nm}(i)$  equals [42]:

$$c_{a,\ell k,nm} = \mathbb{E}\{(a_{\ell k}(i) - \bar{a}_{\ell k})(a_{nm}(i) - \bar{a}_{nm})\} \\ = \begin{cases} c_{a,\ell k,\ell k}, & \text{if } k = m, \ell = n, \ell \in \mathcal{N}_k^- \cap \mathcal{C}(k) \\ -c_{a,\ell k,\ell k}, & \text{if } k = m = n, \ell \in \mathcal{N}_k^- \cap \mathcal{C}(k) \\ -c_{a,nk,nk}, & \text{if } k = m = \ell, n \in \mathcal{N}_k^- \cap \mathcal{C}(k) \\ \sum_{j \in \mathcal{N}_k^- \cap \mathcal{C}(k)} c_{a,jk,jk}, & \text{if } k = m = \ell = n \\ 0, & \text{otherwise.} \end{cases} \quad (93)$$

where  $c_{a,\ell k,\ell k} = a_{\ell k}^2 p_{\ell k} (1 - p_{\ell k})$ .

Finally, the regularization factors  $\{\rho_{k\ell}(i)\}$  are distributed as follows:

$$\rho_{k\ell}(i) = \begin{cases} \rho_{k\ell}, & \text{with probability } r_{k\ell} \\ 0, & \text{with probability } 1 - r_{k\ell} \end{cases} \quad (94)$$

for any  $\ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k)$ , where  $0 < \rho_{k\ell} < 1$  is a fixed regularization factor. The factors  $\{\rho_{k\ell}(i)\}$  are spatially uncorrelated for  $k \neq \ell$ . At each iteration  $i$ , in order to get a right stochastic matrix  $\mathbf{P}(i)$ , node  $k$  adjusts its regularization factor as follows:

$$\rho_{kk}(i) = 1 - \sum_{\ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k)} \rho_{k\ell}(i) \geq 0. \quad (95)$$

The probability distribution (94) allows each agent  $k$  to randomly select a subset of neighbors that do not belong to its cluster and introduce co-regularization in the estimation process. This behavior can also be interpreted as resulting from link random failures between neighboring clusters: at every time instant  $i$ , the communication link from agent  $\ell$  to agent  $k$  drops with probability  $1 - r_{k\ell}$ . The mean of  $\rho_{k\ell}(i)$  is given:

$$\bar{\rho}_{k\ell} = \begin{cases} \rho_{k\ell} r_{k\ell}, & \text{if } \ell \in \mathcal{N}_k \setminus \mathcal{C}(k) \\ 1 - \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} r_{k\ell}, & \text{if } \ell = k \\ 0, & \text{otherwise,} \end{cases} \quad (96)$$

and the covariance between  $\rho_{k\ell}(i)$  and  $\rho_{mn}(i)$  is:

$$c_{\rho,k\ell,mn} = \mathbb{E}\{(\rho_{k\ell}(i) - \bar{\rho}_{k\ell})(\rho_{mn}(i) - \bar{\rho}_{mn})\} \\ = \begin{cases} c_{\rho,k\ell,k\ell}, & \text{if } k = m, \ell = n, \ell \in \mathcal{N}_k \setminus \mathcal{C}(k) \\ -c_{\rho,k\ell,k\ell}, & \text{if } k = m = n, \ell \in \mathcal{N}_k \setminus \mathcal{C}(k) \\ -c_{\rho,kn,kn}, & \text{if } k = m = \ell, n \in \mathcal{N}_k \setminus \mathcal{C}(k) \\ \sum_{j \in \mathcal{N}_k \setminus \mathcal{C}(k)} c_{\rho,kj,kj}, & \text{if } k = m = \ell = n \\ 0, & \text{otherwise} \end{cases} \quad (97)$$

where  $c_{\rho,k\ell,k\ell} = \rho_{k\ell}^2 r_{k\ell} (1 - r_{k\ell})$ .

## APPENDIX B STABILITY OF $\mathcal{F}$

Recall from (68) that

$$\mathcal{F} \approx \mathcal{A}_1^\top [\mathbf{I}_{(NL)^2} - \mathbf{I}_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) - \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) \otimes_b \mathbf{I}_{NL}]. \quad (98)$$

We now upper-bound the spectral radius of  $\mathcal{F}$  in order to derive a sufficient condition for mean-square stability of the algorithm. We can write:

$$\rho(\mathcal{F}) \leq \|\mathcal{A}_1^\top\|_{b,\infty} \cdot \|\mathbf{I}_{(NL)^2} - \mathbf{I}_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) - \mathcal{M}(\mathcal{R}_x + \eta \mathcal{Q}) \otimes_b \mathbf{I}_{NL}\|_{b,\infty} \quad (99)$$

Since the matrix  $\mathcal{A}_1$  is a block left-stochastic matrix, we know that  $\|\mathcal{A}_1^\top\|_{b,\infty} = 1$ . Using (42) and the triangular inequality, we have:

$$\rho(\mathcal{F}) \leq \|\mathbf{I}_{(NL)^2} - \mathbf{I}_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathbf{I}_{NL}) - \mathcal{M}(\mathcal{R}_x + \eta \mathbf{I}_{NL}) \otimes_b \mathbf{I}_{NL}\|_{b,\infty} \\ + \eta \|\mathbf{I}_{NL} \otimes_b \mathcal{M}\mathcal{P}\|_{b,\infty} + \eta \|\mathcal{M}\mathcal{P} \otimes_b \mathbf{I}_{NL}\|_{b,\infty}. \quad (100)$$

Consider the second term on the RHS of (100). We know that

$$\mathbf{I}_{NL} \otimes_b \mathcal{M}\mathcal{P} \stackrel{(54)}{=} (\mathbf{I}_{NL} \otimes_b \mathcal{M})(\mathbf{I}_{NL} \otimes_b \mathcal{P}) \\ \stackrel{(55)}{=} ((\mathbf{I}_N \otimes \bar{\mathbf{M}}) \otimes \mathbf{I}_{L^2}) ((\mathbf{I}_N \otimes \bar{\mathbf{P}}) \otimes \mathbf{I}_{L^2}). \quad (101)$$

Since  $((\mathbf{I}_N \otimes \bar{\mathbf{P}}) \otimes \mathbf{I}_{L^2})$  is a block right-stochastic matrix and  $((\mathbf{I}_N \otimes \bar{\mathbf{M}}) \otimes \mathbf{I}_{L^2})$  is an  $N^2 \times N^2$  block diagonal matrix with each block of the form  $\bar{\mu}_k \mathbf{I}_{L^2}$  ( $k = 1, \dots, N$ ), we obtain:

$$\|\mathbf{I}_{NL} \otimes_b \mathcal{M}\mathcal{P}\|_{b,\infty} \\ \leq \|(\mathbf{I}_N \otimes \bar{\mathbf{M}}) \otimes \mathbf{I}_{L^2}\|_{b,\infty} \cdot \|(\mathbf{I}_N \otimes \bar{\mathbf{P}}) \otimes \mathbf{I}_{L^2}\|_{b,\infty} \\ = \max_{1 \leq k \leq N} \bar{\mu}_k \quad (102)$$

Following the same steps for the third term on the RHS of (100), we have:

$$\|\mathcal{M}\mathcal{P} \otimes_b \mathbf{I}_{NL}\|_{b,\infty} \leq \max_{1 \leq k \leq N} \bar{\mu}_k. \quad (103)$$

The matrix  $[\mathbf{I}_{(NL)^2} - \mathbf{I}_{NL} \otimes_b \mathcal{M}(\mathcal{R}_x + \eta \mathbf{I}_{NL}) - \mathcal{M}(\mathcal{R}_x + \eta \mathbf{I}_{NL}) \otimes_b \mathbf{I}_{NL}]$  in the first term on the RHS of (100) is an  $N^2 \times N^2$  block diagonal matrix. The  $m$ -th block on the diagonal (where  $m = (\ell - 1)N + k$  for  $k, \ell = 1, \dots, N$ ) is of size  $L^2 \times L^2$ , symmetric, and has the following form:

$$\mathbf{I}_{L^2} - \mathbf{I}_L \otimes \bar{\mu}_k (\mathbf{R}_{x,k} + \eta \mathbf{I}_L) - \bar{\mu}_\ell (\mathbf{R}_{x,\ell} + \eta \mathbf{I}_L) \otimes \mathbf{I}_L \\ = (-\bar{\mu}_\ell \mathbf{R}_{x,\ell} - \eta \bar{\mu}_\ell \mathbf{I}_L) \otimes \mathbf{I}_L + \mathbf{I}_L \otimes (\mathbf{I}_L - \bar{\mu}_k \mathbf{R}_{x,k} - \eta \bar{\mu}_k \mathbf{I}_L) \quad (104)$$

Before proceeding, let us recall the Kronecker sum operator, denoted by  $\oplus$ . If  $\mathbf{A}$  and  $\mathbf{B}$  are two matrices of dimension  $L \times L$  each, then

$$\mathbf{A} \oplus \mathbf{B} \triangleq \mathbf{A} \otimes \mathbf{I}_L + \mathbf{I}_L \otimes \mathbf{B}. \quad (105)$$

Let  $\lambda_k\{\cdot\}$  denote the  $k$ -th eigenvalue of its matrix argument. Then, the eigenvalues of  $\mathbf{A} \oplus \mathbf{B}$  are of the form  $\lambda_i\{\mathbf{A}\} +$

$\lambda_j\{\mathbf{B}\}$  for  $i, j = 1, \dots, L$  [50]. Note that the RHS of equation (104) can be written as

$$(-\bar{\mu}_\ell \mathbf{R}_{x,\ell} - \eta \bar{\mu}_\ell \mathbf{I}_L) \oplus (\mathbf{I}_L - \bar{\mu}_k \mathbf{R}_{x,k} - \eta \bar{\mu}_k \mathbf{I}_L) \quad (106)$$

and its eigenvalues are therefore of the form:

$$1 - \eta \bar{\mu}_k - \bar{\mu}_k \lambda_j\{\mathbf{R}_{x,k}\} - \eta \bar{\mu}_\ell - \bar{\mu}_\ell \lambda_i\{\mathbf{R}_{x,\ell}\} \quad (107)$$

for  $i, j = 1, \dots, L$  and  $k, \ell = 1, \dots, N$ . In order to simplify the mean-square stability condition, we assume that the first order moment of the step-sizes is the same for all nodes. Using the fact that the block maximum norm of a block diagonal Hermitian matrix is equal to the largest spectral radius of its block entries [16], we get:

$$\begin{aligned} & \|\mathbf{I}_{(NL)^2} - \mathbf{I}_{NL} \otimes_b \mathcal{M}(\mathbf{R}_x + \eta \mathbf{I}_{NL}) - \\ & \quad \mathcal{M}(\mathbf{R}_x + \eta \mathbf{I}_{NL}) \otimes_b \mathbf{I}_{NL}\|_{b,\infty} \\ &= \max_{1 \leq k, \ell \leq N} \left( \max_{1 \leq i, j \leq L} |1 - 2\eta \bar{\mu} - \bar{\mu}(\lambda_j\{\mathbf{R}_{x,k}\} + \lambda_i\{\mathbf{R}_{x,\ell}\})| \right) \\ &= \max_{1 \leq k, \ell \leq N} \left( \max_{1 \leq i, j \leq L} \{1 - 2\eta \bar{\mu} - \bar{\mu}(\lambda_j\{\mathbf{R}_{x,k}\} + \lambda_i\{\mathbf{R}_{x,\ell}\}), \right. \\ & \quad \left. - 1 + 2\eta \bar{\mu} + \bar{\mu}(\lambda_j\{\mathbf{R}_{x,k}\} + \lambda_i\{\mathbf{R}_{x,\ell}\}) \} \right) \\ &= \max \{1 - 2\eta \bar{\mu} - \bar{\mu} \min_{k,\ell} (\lambda_{\min}\{\mathbf{R}_{x,k}\} + \lambda_{\min}\{\mathbf{R}_{x,\ell}\}), \\ & \quad - 1 + 2\eta \bar{\mu} + \bar{\mu} \max_{k,\ell} (\lambda_{\max}\{\mathbf{R}_{x,k}\} + \lambda_{\max}\{\mathbf{R}_{x,\ell}\}) \}. \end{aligned} \quad (108)$$

The minimum (identically the maximum) on  $k$  and  $\ell$  that appears in the last equality of (108) is reached for  $k = \ell$ . Thus, a sufficient condition for mean-square stability is given by:

$$\max_{1 \leq k \leq N} \left( \max_{1 \leq i \leq L} |1 - 2\eta \bar{\mu} - 2\bar{\mu} \lambda_i(\mathbf{R}_{x,k})| + 2\eta \bar{\mu} \right) < 1, \quad (109)$$

which is verified if the first order moment of the step-sizes satisfies:

$$0 < \bar{\mu} < \frac{1}{2\eta + \max_{1 \leq k \leq N} \rho(\mathbf{R}_{x,k})}. \quad (110)$$

## REFERENCES

- [1] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Performance analysis of multitask diffusion adaptation over asynchronous networks," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 2014.
- [2] A. H. Sayed, "Adaptive networks," *Proc. of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.
- [3] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.
- [4] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Transactions on Automatic Control*, vol. 29, no. 1, pp. 42–50, January 1984.
- [5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *System & Control Letters*, vol. 53, no. 9, pp. 65–78, September 2004.
- [6] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. International Conference on Information Fusion (FUSION)*, Cologne, Germany, June-July 2008, pp. 1–6.
- [7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009.
- [8] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.
- [9] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.
- [10] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, November 1997.
- [11] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, July 2001.
- [12] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal of Selected Topics in Areas in Communications*, vol. 23, no. 4, pp. 798–808, April 2005.
- [13] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, February 2007.
- [14] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, August 2007.
- [15] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.
- [16] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., vol. 3, pp. 322–454. Elsevier, 2014.
- [17] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.
- [18] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.
- [19] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, August 2012.
- [20] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.
- [21] R. Abdolee, B. Champagne, and A. H. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 403–418, Jan. 2014.
- [22] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 7223–7227.
- [23] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," Available as arXiv:1408.3354, 2014.
- [24] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Reims, France, September 2014, pp. 1–6.
- [25] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – Part I: sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.
- [26] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2196–2210, May 2011.
- [27] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, August 2014.
- [28] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion LMS with sparsity-based regularization," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015.
- [29] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.
- [30] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. 3rd International Workshop on Cognitive Information Processing (CIP)*, Baiona, Spain, May 2012, pp. 1–6.
- [31] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, June 2015.

- [32] R. Nassif, C. Richard, J. Chen, A. Ferrari, and A. H. Sayed, "Diffusion LMS over multitask networks with noisy links," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, March 2016.
- [33] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3285–3300, July 2015.
- [34] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [35] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [36] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.
- [37] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, 2008.
- [38] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.
- [39] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1383–1400, 2010.
- [40] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Weight optimization for consensus algorithms with correlated switching topology," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3788–3801, 2010.
- [41] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3889–3902, 2011.
- [42] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part I: Modeling and stability analysis," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 811–826, February 2015.
- [43] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part II: Performance analysis," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 827–842, February 2015.
- [44] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part III: Comparison analysis," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 843–858, February 2015.
- [45] L. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*, Springer, 2010.
- [46] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, London, Academic Press, 2nd edition, 1995.
- [47] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, vol. 33, no. 3, pp. 520–534, 1965.
- [48] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, NJ, 2008.
- [49] R. H. Koning, H. Neudecker, and T. Wansbeek, "Block Kronecker products and the vecb operator," *Linear Algebra and its Applications*, vol. 149, pp. 165–184, April 1991.
- [50] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*, Princeton University Press, 2005.



**Roula Nassif** was born in Beirut, Lebanon in February 1991. She received the bachelor's degree in Electrical Engineering from the Lebanese University, Lebanon, in 2013. She received the M.S. degrees in Industrial Control and Intelligent Systems for Transport from the Lebanese University, Lebanon, and from Compiègne University of Technology, France, in 2013. Since October 2013 she is a Ph.D. student at the Lagrange Laboratory (University of Nice Sophia Antipolis, CNRS, Observatoire de la Côte d'Azur). Her research activity is focused on distributed optimization over multitask networks.





**Cédric Richard** (S'98-M'01-SM'07) received the Dipl.-Ing. and the M.S. degrees in 1994, and the Ph.D. degree in 1998, from Compiègne University of Technology, France, all in electrical and computer engineering. From 1999 to 2003, he was an Associate Professor at Troyes University of Technology, France, and a Full Professor from 2003 to 2009. Since 2009, he is a Full Professor at the University of Nice Sophia Antipolis, France. He was a junior member of the Institut Universitaire de France in 2010-2015.

His current research interests include statistical signal processing and machine learning. Cédric Richard is the author of over 230 papers. He was the General Co-Chair of the IEEE SSP Workshop that was held in Nice, France, in 2011. He was the Technical Co-Chair of EUSIPCO 2015 that was held in Nice, France, and of the IEEE CAMSAP Workshop 2015 that was held in Cancun, Mexico. He serves as a Senior Area Editor of the IEEE Transactions on Signal Processing and as an Associate Editor of the IEEE Transactions on Signal and Information Processing over Networks since 2015. He is also an Associate Editor of Signal Processing Elsevier since 2009. Cédric Richard is member of the Machine Learning for Signal Processing (MLSP TC) Technical Committee, and served as member of the Signal Processing Theory and Methods (SPTM TC) Technical Committee in 2009-2014.



**André Ferrari** (SM'91-M'93) received the Ingénieur degree from École Centrale de Lyon, Lyon, France, in 1988 and the M.Sc. and Ph.D. degrees from the University of Nice Sophia Antipolis (UNS), France, in 1989 and 1992, respectively, all in electrical and computer engineering. He is currently a Professor at UNS.

He is currently a Professor at UNS. He is a member of the Joseph-Louis Lagrange Laboratory (CNRS, OCA), where his research activity is centered around statistical signal processing and modeling, with a particular interest in applications to astrophysics.



**Ali H. Sayed** (S'90-M'92-SM'99-F'01) is professor and former chairman of electrical engineering at the University of California, Los Angeles, USA, where he directs the UCLA Adaptive Systems Laboratory. An author of more than 460 scholarly publications and six books, his research involves several areas including adaptation and learning, statistical signal processing, distributed processing, network science, and biologically inspired designs. Dr. Sayed has received several awards including the 2015 Education Award from the IEEE Signal Processing Society, the

2014 Athanasios Papoulis Award from the European Association for Signal Processing, the 2013 Meritorious Service Award, and the 2012 Technical Achievement Award from the IEEE Signal Processing Society. Also, the 2005 Terman Award from the American Society for Engineering Education, the 2003 Kuwait Prize, and the 1996 IEEE Donald G. Fink Prize. He served as Distinguished Lecturer for the IEEE Signal Processing Society in 2005 and as Editor-in Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2003-2005). His articles received several Best Paper Awards from the IEEE Signal Processing Society (2002, 2005, 2012, 2014). He is a Fellow of the American Association for the Advancement of Science (AAAS). He is recognized as a Highly Cited Researcher by Thomson Reuters.



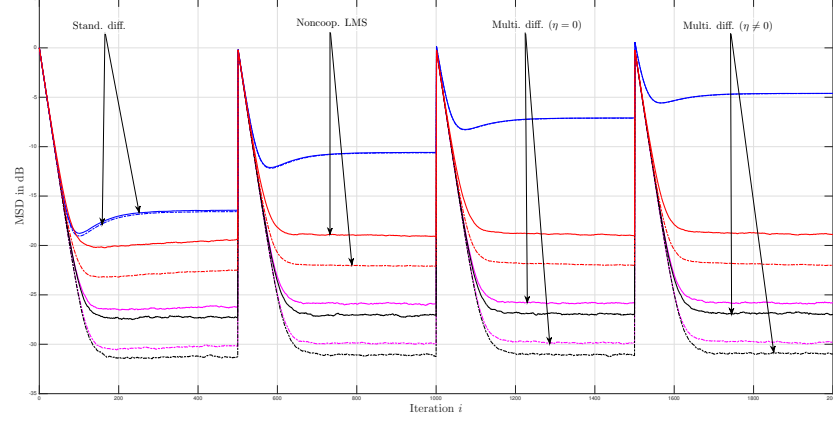


Fig. 7. Network topology consisting of 10 clusters. Network MSD learning curves in a non-stationary environment: comparison of the multitask diffusion LMS with (namely,  $\eta > 0$ ) and without (namely,  $\eta = 0$ ) inter-cluster cooperation, the standard diffusion LMS [17] and the non-cooperative LMS. The dotted lines are for synchronous networks and the solid lines are for asynchronous networks.

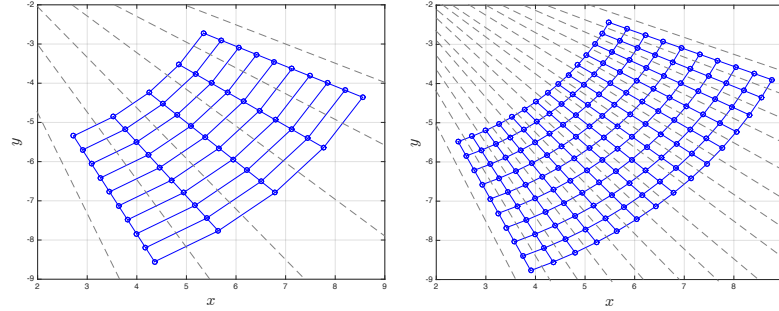


Fig. 9. Network topology: circles for nodes, solid lines for links, and dashed lines for cluster boundaries. Left: network consisting of 5 clusters. Right: network consisting of 15 clusters.

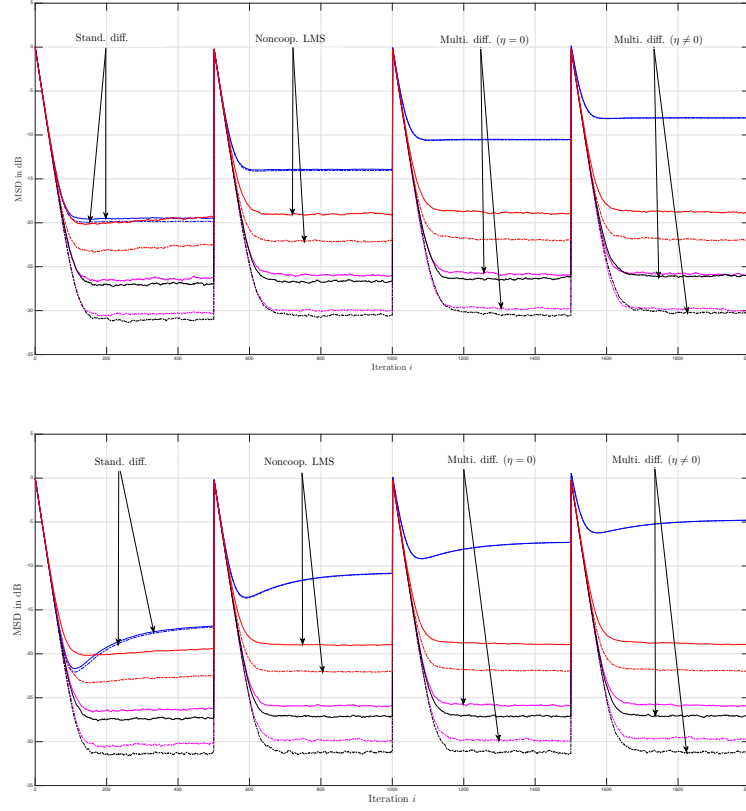


Fig. 10. Network MSD learning curves in a non-stationary environment: comparison of the same algorithms considered in Figure 7. The dotted lines are for synchronous networks and the solid lines are for asynchronous networks. Top: network consisting of 5 clusters. Down: Network consisting of 15 clusters.